

Université de Montréal

Détection des chutes par calcul homographique

par

Djamila Mokhtari

Département d'Informatique et de Recherche Opérationnelle

Faculté des Arts et des Sciences

Mémoire présenté à la Faculté des Arts et des Sciences

en vue de l'obtention du grade de Maîtrise

en Informatique

Août, 2012

© Djamila Mokhtari, 2012

Résumé

La vidéosurveillance a pour objectif principal de protéger les personnes et les biens en détectant tout comportement anormal. Ceci ne serait possible sans la détection de mouvement dans l'image. Ce processus complexe se base le plus souvent sur une opération de soustraction de l'arrière-plan statique d'une scène sur l'image. Mais il se trouve qu'en vidéosurveillance, des caméras sont souvent en mouvement, engendrant ainsi, un changement significatif de l'arrière-plan; la soustraction de l'arrière-plan devient alors problématique. Nous proposons dans ce travail, une méthode de détection de mouvement et particulièrement de chutes qui s'affranchit de la soustraction de l'arrière-plan et exploite la rotation de la caméra dans la détection du mouvement en utilisant le calcul homographique. Nos résultats sur des données synthétiques et réelles démontrent la faisabilité de cette approche.

Mots-clés : Rotation de caméra, points SIFT, points SURF, mise en correspondance, homographie, détection de mouvement, détection de chute.

Abstract

The main objective of video surveillance is to protect persons and property by detecting any abnormal behavior. This is not possible without detecting motion in the image. This process is often based on the concept of subtraction of the scene background. However in video tracking, the cameras are themselves often in motion, causing a significant change of the background. So, background subtraction techniques become problematic. We propose in this work a motion detection approach, with the example application of fall detection. This approach is free of background subtraction for a rotating surveillance camera. The method uses the camera rotation to detect motion by using homographic calculation. Our results on synthetic and real video sequences demonstrate the feasibility of this approach.

Keywords : Camera rotation, SIFT keypoints, SURF keypoints, matching, homography, motion detection , fall detection.

Table des matières

Liste des tableaux.....	vii
Liste des figures.....	viii
Liste des algorithmes	x
Liste des sigles et abréviations.....	xi
Introduction générale	1
Chapitre 1: Étude bibliographique	3
1- La vidéosurveillance.....	3
1-1- Objectifs de la vidéosurveillance.....	3
1-2- Les domaines d'application.....	3
1-3- Quelques projets de vidéosurveillance.....	4
1-4- Structure d'un système de vidéosurveillance.....	5
2- La détection du mouvement.....	6
2-1- Méthodes basées sur la modélisation de l'arrière-plan.....	7
2-1-1- Utilisation de l'image de l'arrière-plan.....	8
2-1-2- Modélisation de l'image de l'arrière-plan.....	8
2-1-2-1- Méthode de la moyenne temporelle.....	9
2-1-2-2- Méthode de distribution gaussienne simple.....	9
2-1-2-3- Méthode de mixture de distributions gaussiennes....	11
2-1-2-4- Méthode Min, Max et écart Max.....	12
2-1-2-5- Méthode basée sur la texture.....	13
2-2- Méthodes basées sur la mesure du mouvement.....	14
2-2-1- Méthode différentielle	14
2-2-2- Le flux optique.....	15
2-3- Les méthodes hybrides.....	17
2-4-Élimination des ombres.....	18
2-4-1- Seuillage en HSV.....	18
2-4-2- Méthode statistique.....	19
2-4-3- Méthode du gradient et chromaticité.....	20

3- Le suivi des objets.....	20
3-1-Méthode de recouvrement.....	21
3-2- Utilisation du filtre de Kalman.....	21
3-3- Utilisation du filtre à particules.....	22
4- La reconnaissance.....	22
4-1- Prise de médicaments.....	23
4-2- Reconnaissance de démarches.....	24
5- La vidéosurveillance appliquée à la détection des chutes.....	25
5-1- Intérêts et objectifs.....	25
5-2- Caractéristiques d'une chute.....	26
5-3- Méthodes de détection de chutes par vidéosurveillance.....	27
5-4- Méthodes de détection de chutes sans vidéosurveillance.....	28
6- Conclusion.....	29
Chapitre 2: Aspects théoriques de l'algorithme de détection de chutes.	30
1- Introduction.....	30
2- Problématique et objectif.....	30
3- Intérêts du mouvement de rotation de la caméra.....	32
4- Modélisation de la caméra.....	33
4-1- Projection d'un point 3D sur l'image.....	35
5- Extraction des indices visuels de l'image.....	40
5-1- Détection des points d'intérêts.....	41
5-2- Calcul des descripteurs.....	42
6- Mise en correspondance de deux images.....	44
6-1- Stratégies de mise en correspondance.....	46
7- Calcul homographique.....	46
7-1- Estimation de l'homographie.....	49
7-1-1- L'algorithme DLT.....	49
7-1-2- Normalisation des pixels.....	50
7-1-3-Optimisation de l'estimation de l'homographie par Ransac.....	51
7-1-4- Calcul du nombre d'itérations.....	52

8- Détection de mouvement.....	54
8-1- Calcul des points en mouvement.....	54
8-2- Élimination des faux positifs.....	55
9- Détection de chute.....	58
10- Conclusion.....	60
Chapitre 3: Expérimentation.....	61
1- Introduction.....	61
2- Environnement de l'expérimentation.....	61
2-1- Configuration de la machine.....	61
2-2- Langage de programmation.....	61
3- Données de l'expérimentation.....	62
3-1- Images de synthèse.....	62
3-2- Images réelles.....	62
4- Description du processus de réalisation.....	63
4-1- Génération des Images de synthèse / Acquisition des images réelles....	64
4-2- Extraction des points d'intérêt.....	64
4-3- Mise en correspondance des points d'intérêt.....	64
4-4- Calcul de l'homographie.....	64
4-5- Détection des points en mouvements.....	64
4-6- Classification du mouvement.....	64
5- Images de synthèse : Traitement et interprétation des résultats.....	65
5-1- Génération des Images de synthèse.....	65
5-2- Extraction des points d'intérêts.....	67
5-3- Mise en correspondance des points d'intérêts.....	68
5-4- Calcul de l'homographie.....	75
5-5- Détection des points en mouvements.....	75
5-6- Détection de chutes.....	83
5-7- Conclusion sur le traitement des images de synthèse.....	90
6- Images réelles : Traitement et interprétation des résultats.....	91
6-1- Détection de mouvement.....	91

6-2- Détection de Chute.....	94
6-3- Discussion.....	97
Conclusion générale.....	98
Références bibliographiques.....	100

Liste des tableaux

	Page
Tableau I : Nombre et importance relative des aînés, selon l'âge, Québec, 1981 à 2041	25

Liste des figures

	Page
Figure 1.1 : Système de vidéosurveillance.....	6
Figure 1.2 : Détection de mouvement par modélisation de l'arrière-plan.....	7
Figure 1.3 : Processus de détection de mouvement.....	7
Figure 1.4 : Détection de mouvement par différence temporelle.....	15
Figure 1.5 : Le flux optique.....	15
Figure 1.6 : Détection de mouvement par méthode hybride.....	18
Figure 1.7 : Calcul des distorsions et volume d'ombre.....	19
Figure 1.8 : Le cycle du filtre de Kalman.....	21
Figure 1.9 : Nombre de pixels des jambes en mouvement.....	24
Figure 1.10 : Les quatre phases d'une chute.....	26
Figure 2.1 : Disposition de la caméra dans la pièce.....	31
Figure 2.2 : Objet en mouvement sur deux images prises par une caméra en rotation pure.....	31
Figure 2.3 : Modélisation de la caméra et projection d'un point 3D.....	33
Figure 2.4. : Angle de vue de la caméra.....	34
Figure 2.5 : Projection perspective d'un point du monde par changement de repère	35
Figure 2.6. : Projection d'un point P sur le plan image.....	38
Figure 2.7: Calcul des points SIFT : Pyramide des calculs à 2 octaves.....	41
Figure 2.8 : Calcul du descripteur d'un point SIFT.....	42
Figure 2.9 : Invariance des points SIFT à la translation, aux changements d'échelle et à la rotation pour la construction d'un panorama.....	43
Figure 2.10 : Mise en correspondance de deux points images.....	44
Figure 2.11 : Effet de la transformation perspective sur la taille du nuage de points	59
Figure 3.1 : Synoptique des différents traitements.....	63
Figure 3.2 : Calcul géométrique des paramètres de génération des images de synthèse	65
Figure 3.3: Image de la séquence vidéo synthétisée.....	67
Figure 3.4 : Construction de l'ensemble de validation.....	68

Figure 3.5 : Variation du Nombre de points d'intérêts (Key) et du nombre de points mis en correspondance (Crsp).....	71
Figure 3.6 : Nombre de points d'intérêts non appariés en fonction de l'angle de rotation θ	72
Figure 3.7 : Mise en correspondance de deux images successives.	74
Figure 3.8 : Résultat de la détection du mouvement pour différentes valeurs de θ .	78
Figure 3.9 : Variation du nombre moyen de points détectés en fonction de θ .	79
Figure 3.10 : Pourcentage de points détectés en mouvement en fonction de l'angle θ .	80
Figure 3.11 : Détection du mouvement.	82
Figure 3.12 : Séquence de 30 images avec mouvement.	84
Figure 3.13 : Variation du nombre de chutes détectées.	85
Figure 3.14 : Variation de la F_{mesure} en fonction des valeurs de a	86
Figure 3.15 : Variation du seuil ρ et de la distance $d(n_1, n_2)$, pour $a=10$.	87
Figure 3.16 : Résultat du détecteur de chute (algorithme 2.5).	89
Figure 3.17 : Séquence d'images avec déplacement du robot et rotation de la Caméra	92
Figure 3.18: Détection des points en mouvement.	93
Figure 3.19 : Séquence de chute du robot.	96

Liste des algorithmes

	Page
Algorithme 2.1 : Calcul de H par DLT avec normalisation des pixels.....	51
Algorithme 2.2 : Estimation de l'homographie par RANSAC adaptatif.....	53
Algorithme 2.3 : Détection des points en mouvement.....	54
Algorithme 2.4 : Amélioration de la détection des points en mouvement.....	57
Algorithme 2.5 : Détection de chute.....	60
Algorithme 3.1 : Génération des images de synthèse.....	67

Liste des sigles et des abréviations

2D	Deux dimensions
3D	Trois dimensions
<i>B</i>	Image arrière-plan (Background)
BLOB	Binary Large Object
Crsp	Nombre de points mis en correspondance
DLT	Direct Linear Transform
HSV	Hue Saturation Value (Teinte Saturation Intensité)
<i>I</i>	Image
Key	Nombre de points d'intérêts
LBP	Local Binary Pattern
MC	Nombre de points en mouvement mis en correspondance
MD	Nombre de points en mouvement détectés
RGB	Red Green Blue
SC	Nombre de points statiques mis en correspondance
SD	Nombre de points statiques détectés
SIFT	Scale Invariant Features Transform
SURF	Speeded Up Robust Features
SVD	Singular Value Decomposition
VSAM	Video Monitoring and Activity Monitoring
W ⁴	Who? When? Where? What?

Remerciements

Je voudrais remercier très sincèrement, tous ceux qui de près ou de loin ont contribué à l'élaboration de ce travail. Je souhaiterais particulièrement faire part de ma gratitude à :

Mon directeur de recherche, le Pr. Jean Meunier, du DIRO de l'Université de Montréal, qui n'a pas lésiné sur son temps et ses efforts pour m'apporter les orientations et les corrections nécessaires à la réalisation de ce travail.

Messieurs les Professeurs Neil Stewart et Max Mignotte qui me font l'honneur d'évaluer ce travail.

Mes collègues du laboratoire de vision et traitement d'images pour leur soutien, particulièrement Edouard Auvinet et Sébastien Roy.

Aux membres de ma famille, qu'ils me pardonnent de les remercier en dernier, et tout particulièrement ma mère et mon époux, qui par leur soutien permanent et leurs encouragements ont créé un climat favorable à la réalisation de ce travail.

INTRODUCTION GÉNÉRALE

La vidéosurveillance a pour objectif primordial, la sécurité et la protection des personnes et des biens. Ceci est réalisable en analysant constamment des flots vidéos afin de déceler tout comportement anormal ou suspect. Mais la quantité importante des données nécessite une automatisation du processus d'analyse. Ainsi nous assistons à des recherches laborieuses qui tentent de doter les ordinateurs d'algorithmes puissants permettant de distinguer un comportement normal d'un comportement anormal et par suite générer une alerte.

La vidéosurveillance est appliquée dans différents domaines. Nous nous intéressons dans ce mémoire à un domaine particulier qu'est la détection de chutes auprès d'une population de plus en plus vieillissante.

Ce travail s'inscrit dans le cadre d'un projet de recherche pour la détection des chutes, au sein du laboratoire "Vision et traitement d'images" du département d'informatique et de recherche opérationnelle de l'Université de Montréal. Ce projet est scindé en plusieurs sujets qui mettent en œuvre différents concepts et modèles mathématiques de détection de chutes.

Le sujet de ce présent mémoire consiste à utiliser le calcul homographique dans le but de détecter du mouvement et plus particulièrement un mouvement de chute entre deux images successives d'une séquence d'images prises par une caméra en rotation pure.

Ce mémoire est scindé en trois chapitres. Le premier est consacré à une étude bibliographique, où nous décrivons différentes méthodes de détection de mouvement, de suivi et de reconnaissance.

Dans le second chapitre, nous présentons les aspects théoriques de la méthode développée. Le troisième chapitre fait l'objet de l'expérimentation de cette méthode. Enfin nous terminons par une conclusion.

Chapitre 1

ÉTUDE BIBLIOGRAPHIQUE

1- La vidéosurveillance

La vidéosurveillance est un système de caméras disposées dans un espace à surveiller. Ces caméras sont reliées à un système informatique qui permet le traitement et l'analyse des données reçues. Le premier système de vidéosurveillance a été conçu en Allemagne en 1942, par Siemens AG pour des fins d'observations de fusées [4]. Depuis, les systèmes de vidéo surveillance ont considérablement évolué. L'analyse et l'intégration des données sont de plus en plus automatisées et suscitent moins d'intervention humaine.

1-1- Objectifs de la vidéosurveillance

La vidéosurveillance a pour but de protéger les personnes, les biens, et les espaces privés ou publics. Elle a aussi un rôle rassurant auprès des personnes vulnérables et un rôle dissuasif auprès des personnes potentiellement criminelles. En plus, la vidéosurveillance s'est intégrée, ces dernières années, dans les systèmes de maison intelligente [64].

1-2- Les domaines d'application

Les domaines d'application de la vidéosurveillance sont variés et peuvent être publics ou privés. On peut citer :

- La surveillance et la protection dans les espaces publics (aéroport, métro, centres commerciaux...);
- la gestion des espaces [1]: visualisation des places libres dans un port, dans un parking...;
- l'utilisation touristique [1]: fournir de l'information au public via des sites internet;

- la surveillance du trafic routier ;
- la sécurité des habitations;
- la reconnaissance d'activité;
- la reconnaissance de mouvements;
- l'analyse de prise de médicaments [35],[60]
- l'analyse de prise des repas [30],
- la reconnaissance d'expressions faciales [23],
- la détection de chutes [52],
- la détection des objets abandonnés [10],
- la détection d'événements suspects [40],
- la maison intelligente [64].

1-3- Quelques projets de vidéosurveillance

Le projet **VSAM** (Video Surveillance and Activity Monitoring) [18] est un système de vidéosurveillance développé en 2000, par le *Robotics Institute* de *Carnegie Mellon University* (CMU) et la *Sarnoff Corporation*. Son objectif est la détection et le suivi automatique d'objets mobiles tels que des personnes et des véhicules dans un environnement urbain, en utilisant un réseau de caméras distribuées. Cette approche est basée sur la soustraction de l'arrière plan. En utilisant un système de réseaux de neurones, les objets qui sont en mouvements sont d'abord classés par catégorie (humain, groupes et voitures), ensuite classés selon leurs activités (courir, marcher, descendre d'un véhicule...).

Le projet **PROSAFE** [17] identifie les comportements anormaux qui peuvent être interprétés comme un accident basée sur des données de capteurs et de résultats puis génère des alarmes d'urgence.

Le projet **ADVISOR** [55] [20] (Annotated Digital Video for Surveillance and Optimised Retrieval) est un projet européen de l'équipe ORION de l'INRIA pour la surveillance des stations de métro. Le principal objectif de ce système est de sécuriser les transports publics en

détectant automatiquement toute situation anormale pouvant induire des accidents, de la violence ou des actes de vandalisme [39].

La détection des objets en mouvement dans la scène se fait par la soustraction de l'arrière-plan qui est mis à jour périodiquement. Les objets mobiles extraits sont décrits par le centre de gravité, la position, etc. et par une classe sémantique (ex. personne). Le suivi des objets mobiles se fait en reliant les objets mobiles sur deux images successives en tenant compte de la similitude, la distance ainsi que la classe sémantique. Cela fournit un graphe des objets mobiles avec les trajectoires possibles de chaque objet mobile. La reconnaissance des comportements se fait en associant la sémantique des objets mobiles aux zones d'intérêt tels que : barrières de passage, distributeur de ticket...

Le projet W⁴ [31] est un système de vidéosurveillance temps réel pour la détection et le suivi de multiples personnes à l'extérieur. Il opère à l'aide d'une monocaméra monochrome ou infrarouge. Il combine l'analyse des silhouettes des personnes et le suivi des personnes afin de localiser les différentes parties du corps (tête, bras, pieds...).

Le projet **BEHAVE** : Le but du projet est la détection des comportements anormaux et/ou criminels [9]. Il utilise les modèles de Markov cachés dynamiques pour la reconnaissance des comportements des personnes.

Le projet **AVITRACK** [28] : L'objectif du projet est l'identification des activités autour des avions au sol.

1-4- Structure d'un système de vidéosurveillance

Un système de vidéosurveillance est composé de 3 modules :

Le module de détection : La détection est la première étape d'un système de vidéosurveillance. Elle permet l'extraction des objets mobiles présents sur les images de la séquence vidéo en les séparant de l'arrière-plan. C'est une étape de traitement de bas niveau.

Le module de suivi : Le suivi des objets mobiles est une estimation temporelle des trajectoires des différents objets mobiles.

Le module de reconnaissance: Il permet de reconnaître des comportements suspects et génère une prise de décision appropriée à ce comportement.

La figure suivante illustre les trois modules :

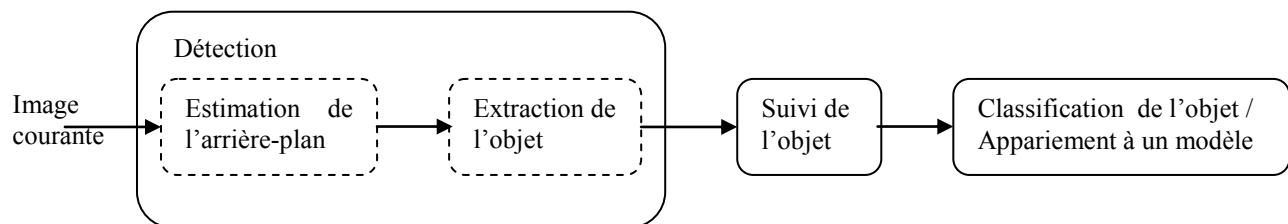


Figure 1.1 : Système de vidéosurveillance [21].

2- La détection du mouvement

La détection du mouvement constitue la première étape d'un système de vidéosurveillance, elle permet de détecter les objets mobiles sur la scène. C'est une étape critique et difficile car elle doit être robuste aux variations de la luminosité de la scène et la présence des ombres. Il existe différentes méthodes de détection du mouvement que l'on peut regrouper en trois grandes classes [18] :

- Les méthodes basées sur la modélisation de l'arrière-plan et la mesure de différence entre une image de la séquence et celle de l'arrière-plan.
- Les méthodes basées sur des mesures des zones de mouvement.
- Les méthodes hybrides combinant les deux principes précédents.

2-1- Méthodes basées sur la modélisation de l'arrière-plan

Plusieurs travaux sur la détection d'objets mobiles, procèdent par une opération de segmentation qui permet de soustraire l'arrière-plan d'une image et ainsi détecter les objets mobiles pour déterminer leur trajectoire lors de l'étape de suivi (Figure 1.2).



Figure 1.2: Détection de mouvement par modélisation de l'arrière-plan.

(a) Scène avec mouvement, (b) Extraction des objets mobiles par soustraction de l'arrière-plan (adapté de [2]).

Plusieurs méthodes de détection de mouvement par soustraction d'arrière-plan ont été proposées en littérature. Elles présentent pour la plupart un modèle de l'arrière-plan qui est mis à jour. La figure 1.3 illustre le processus de détection du mouvement par la mise à jour de l'arrière-plan :

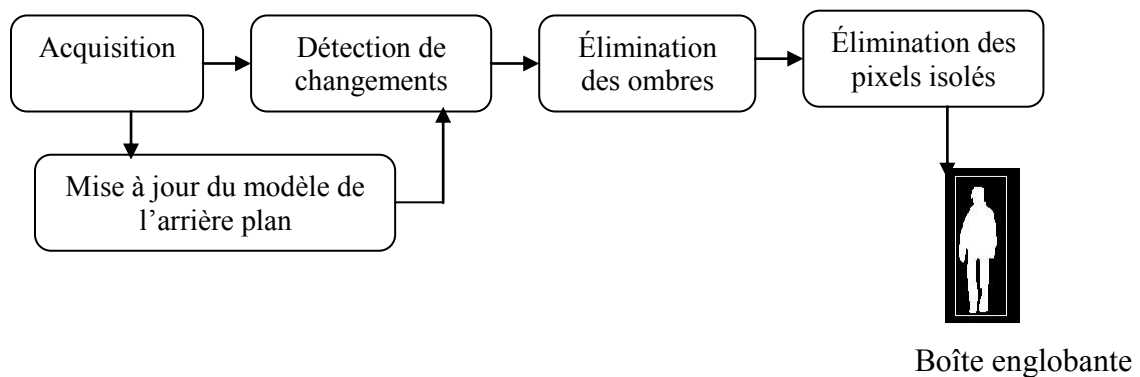


Figure 1.3 : Processus de détection de mouvement (adaptée de [21]).

La mise-à-jour de l'arrière-plan se fait au fur et à mesure de l'acquisition des images, une opération de détection des changements au niveau des pixels permet d'extraire les objets mobiles, suivent alors des opérations de post-traitements afin d'éliminer les ombres des objets mobiles et les pixels isolés résultant des différents traitements sur l'image. Enfin, les pixels appartenant à l'avant-plan sont regroupés dans des boîtes englobantes (BLOB : Binary Large Object), délimitant ainsi les différentes silhouettes des objets mobiles.

2-1-1- Utilisation de l'image de l'arrière-plan

L'image de l'arrière-plan d'une scène contient les éléments statiques de cette scène, c'est la représentation de l'environnement dans lequel les objets évoluent [21]. Cette image est considérée comme une image de référence qui se soustrait de chaque image de la séquence afin d'extraire les objets en mouvement.

Cette méthode dans sa version simple, consiste à faire la différence des intensités des pixels $I(x,y,t)$ d'une image I prise à l'instant t , par rapport aux intensités des pixels $B(x,y)$ d'une image de référence d'arrière-plan B , où il n'y a pas de mouvement. Elle permet de détecter les pixels (x,y) ayant subi un changement d'intensité significatif. Ces pixels forment les objets en mouvement. Cela est modélisé par :

Si $|I(x,y,t) - B(x,y)| > \tau$ alors le pixel (x,y) de I à l'instant t est en mouvement
sinon le pixel (x,y) appartient à l'arrière-plan

avec τ un seuil fixé manuellement.

Le seuillage rend cette méthode peu robuste et les problèmes liés au bruit et aux changements de la luminosité demeurent présents dans cette méthode.

2-1-2- Modélisation de l'image de l'arrière-plan

L'arrière-plan est non uniforme à travers la séquence d'images, à cause du bruit du capteur et la présence de petits mouvements dans l'arrière-plan ainsi que des changements

graduels de la luminosité. Pour pallier à cette non-uniformité, plusieurs méthodes de modélisation de l'arrière-plan ont été proposées.

2-1-2-1- Méthode de la moyenne temporelle

Cette méthode consiste à calculer un modèle d'arrière-plan qui est continuellement mis à jour. Initialement on calcule l'image moyenne temporelle à partir de M premières images de la séquence. Cette image moyenne sera considérée comme l'arrière-plan initial de la scène :

$$B(x, y, M) = \frac{1}{M} \sum_{t=1}^M I(x, y, t)$$

où $I(x, y, t)$: est la valeur du pixel (x, y) , sur l'image I au temps t et

$B(x, y, t)$: est la valeur du pixel (x, y) , sur l'image arrière-plan au temps t .

L'arrière-plan est continuellement mis à jour pour les pixels appartenant à l'arrière-plan dans l'image courante. On considère que chaque image de la séquence contribue à la mise à jour de l'arrière plan:

$$B(x, y, t) = \frac{(t-1)B(x, y, t-1) + I(x, y, t)}{t}$$

On considère qu'il y a du mouvement si:

$$|I(x, y, t) - B(x, y, t)| > \tau$$

où τ est un seuil fixé manuellement.

2-1-2-2- Méthode de distribution gaussienne simple

McKenna et al [44] proposent une méthode adaptative, où l'arrière-plan est mis à jour par l'information contenue dans chaque image de la séquence, c'est-à-dire que chaque image de la séquence contribue à la formation de l'arrière-plan par ses pixels qui ne sont pas en

mouvement, par rapport à l'image précédente. Cette méthode prend en considération le bruit du capteur qui rend l'arrière-plan non uniforme, ainsi que les changements graduels de la luminosité de la scène.

Le bruit du capteur est considéré comme un bruit gaussien de paramètres estimés $\sigma^2_{\text{cam}} = (\sigma^2_{\text{rcam}}, \sigma^2_{\text{gcam}}, \sigma^2_{\text{bcam}})$, selon les 3 canaux RGB. De plus, chaque pixel de l'image de référence (c-à-d arrière-plan) est modélisé sous forme d'une distribution gaussienne afin de tenir compte du bruit et l'image de référence est mise à jour afin de tenir compte des changements graduels de la luminosité.

Les différentes valeurs que prend un pixel (x,y) dans le temps constituent un processus X : $X = \{X_1, \dots, X_T\} = \{I(x,y,t) : 1 \leq t \leq T\}$

Ce processus est modélisé pour chaque pixel par une distribution gaussienne de moyenne $\mu_t = (\mu_{rt}, \mu_{gt}, \mu_{bt})$ pour les 3 canaux RGB et de variance $\sigma^2_t = (\sigma^2_{rt}, \sigma^2_{gt}, \sigma^2_{bt})$:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)}$$

où n : nombre de canaux (généralement $n=3$ pour RGB) et

$\Sigma = \sigma \mathbf{I}$ (où \mathbf{I} est la matrice identité).

Ces paramètres sont mis à jour selon les équations suivantes :

$$\mu_{t+1} = \alpha \mu_t + (1-\alpha) X_{t+1}$$

$$\sigma^2_{t+1} = \alpha (\sigma^2_t + (\mu_{t+1} - \mu_t)^2) + (1-\alpha) (X_{t+1} - \mu_{t+1})^2$$

où X_{t+1} est la valeur du pixel $I(x,y,t+1)$ et

α est un paramètre de contrôle du taux d'adaptation ($0 < \alpha < 1$).

Un pixel est classé en mouvement s'il satisfait à au moins une des conditions suivantes :

$$|X_{rt} - \mu_{rt}| > 3 \max(\sigma_{rt}, \sigma_{\text{rcam}})$$

$$|X_{gt} - \mu_{gt}| > 3 \max(\sigma_{gt}, \sigma_{\text{gcam}})$$

$$|X_{bt} - \mu_{bt}| > 3 \max(\sigma_{bt}, \sigma_{\text{bcam}})$$

Les valeurs σ_{cam} sont initialisées à une petite valeur correspondant à un faible bruit qui peut être dû à des vibrations de la caméra (qui est fixe). La valeur 3 est choisie, car les valeurs d'une gaussienne sont centrées autour de sa moyenne et sont dans un intervalle $[-3\sigma, 3\sigma]$ (σ étant l'écart type de cette gaussienne).

2-1-2-3- Méthode de mixture de distributions gaussiennes

Stauffer et al [58] proposent une méthode qui considère que l'arrière-plan n'est pas uniforme à cause du bruit du capteur ou de la présence de petits mouvements (par exemple : feuillage d'un arbre) et qu'il y a des changements graduels de la luminosité.

Chaque pixel de l'image de référence est modélisé sous forme d'un mélange de distributions gaussiennes afin de tenir compte du bruit et des changements de l'arrière plan. L'image de référence est mise à jour afin de tenir compte des changements de luminosité.

Les différentes valeurs que prend un pixel (x,y) dans le temps constituent un processus X : $X = \{X_1, \dots, X_T\} = \{I(x,y,t) : 1 \leq t \leq T\}$

Ce processus est modélisé par un mélange de K distributions gaussiennes. Ainsi, la probabilité d'avoir une intensité X_t est donnée par:

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

$\omega_{i,t}$ est le poids de la $i^{\text{ème}}$ distribution gaussienne du mélange de distributions à l'instant t ; cela représente la proportion des données utilisées dans le calcul de la gaussienne.

$\mu_{i,t}$ est la moyenne de la $i^{\text{ème}}$ distribution gaussienne du mélange de distributions à l'instant t et $\Sigma_{i,t}$ est sa matrice de covariance avec $\Sigma_{i,t} = \sigma_i^2 \mathbf{I}$ (\mathbf{I} est la matrice identité).

η représente la fonction de densité de probabilité d'une gaussienne i :

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{n/2} |\Sigma_{i,t}|^{1/2}} e^{-\frac{1}{2}(X_t - \mu_{i,t})^T \Sigma_{i,t}^{-1} (X_t - \mu_{i,t})}$$

où n : nombre de canaux (généralement $n=3$ pour RGB),

$\mu_{i,t}=(\mu_{i,rt}, \mu_{i,gt}, \mu_{i,bt})$ est la moyenne de la gaussienne i à l'instant t , pour les 3 canaux RGB,

$\sigma_{i,t}^2=(\sigma_{i,rt}^2, \sigma_{i,gt}^2, \sigma_{i,bt}^2)$ sa variance et

$\Sigma_{i,t} = \sigma_{i,t} \cdot \mathbf{I}$ est la matrice de covariance de la gaussienne i à l'instant t , avec \mathbf{I} la matrice identité.

Un pixel X_t provient d'un objet en mouvement si:

$$|X_t - \mu_{i,t}| > 3\sigma_{i,t} \quad \text{pour les } S \text{ distributions}$$

avec $S = \operatorname{argmin}_b \left(\sum_{k=1}^b \omega_k > \rho \right)$, où ρ est la proportion minimum de données choisies dans la modélisation de l'arrière-plan.

2-1-2-4- Méthode Min, Max et écart Max

Haritaoglu et al [31] présentent une méthode qui se base sur la valeur maximale et minimale que prend un pixel ainsi que sur l'écart maximal de la valeur d'un pixel dans deux images successives, pour modéliser l'arrière-plan.

La médiane des images est calculée pour les 20 à 40 secondes des premières images de la séquence. Chaque pixel X_t est caractérisé au temps t , par les paramètres $\min(X_t)$, $\max(X_t)$ et $d(X_t)$, où $d(X_t)$ est l'écart maximal du pixel x entre 2 images consécutives, sur N images successives.

Un pixel X_t provient d'un objet statique si:

$$|X_t - \min(X_t)| < k d_\mu \quad \text{ou} \quad |X_t - \max(X_t)| < k d_\mu$$

où d_μ est la médiane des $d(X_t)$ dans l'image complète et

k est une constante seuil déterminée par expérimentation, l'auteur propose $k=2$.

2-1-2-5- Méthode basée sur la texture

Contrairement aux méthodes présentées ci-dessus qui opèrent sur chaque pixel, Heikkilä et al [34] présentent une méthode qui considère un groupe de pixels pour obtenir une plus grande robustesse aux changements d'intensité et d'illumination. Cette méthode utilise une description de texture avec LBP (patrons locaux binaires), avec un seuillage du voisinage d'un pixel par la valeur de celui-ci, pour des images à niveaux de gris:

$$\text{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p$$

$$\text{avec} \quad \begin{aligned} s(x) &= 1 \quad \text{si} \quad x \geq 0 \\ s(x) &= 0 \quad \text{si} \quad x < 0 \end{aligned}$$

g_c est la valeur d'intensité du pixel central (x_c, y_c)

g_p valeur d'intensité du pixel p des P pixels disposés à égale distance sur un cercle de rayon R .

L'arrière-plan est modélisé par calcul d'histogramme. Initialement, on calcule pour chaque pixel, K histogrammes des LBP $\{m_0, \dots, m_{K-1}\}$ sur K régions circulaires autour du pixel, K étant une valeur choisie. Chaque histogramme a un poids ω_k entre 0 et 1, de telle façon que la somme des poids des K histogrammes soit égale à 1. Le modèle d'arrière-plan pour un pixel est un ensemble adaptatif des K histogrammes.

Pour chaque pixel d'une image donnée, on calcule l'histogramme h des LBP puis on le compare aux K histogrammes du modèle. Si l'histogramme calculé h ne ressemble à aucun des K histogrammes modèles, celui-ci remplace l'histogramme modèle ayant le poids le plus faible et un poids faible lui est assigné.

S'il y a correspondance avec un histogramme modèle, celui-ci est mis à jour par :

$$m_k = \alpha_b h + (1 - \alpha_b) m_k$$

où $\alpha_b \in [0,1]$ est un coefficient déterminé par expérimentation.

Les poids attribués aux histogrammes sont également mis à jour par :

$$\omega_k = \alpha_{\omega} M_k + (1 - \alpha_{\omega}) \omega_k$$

où $\alpha_{\omega} \in [0,1]$ est un coefficient déterminé par expérimentation.

Un pixel d'une image est classé en avant-plan si son histogramme h est différent des histogrammes modèles.

2-2- Méthodes basées sur la mesure du mouvement

2-2-1- Méthode différentielle

La présence de mouvement dans une image, peut se définir comme un changement de la valeur de l'intensité d'au moins un pixel entre deux images consécutives dans une séquence vidéo. La différence de deux images successives devrait permettre de détecter un mouvement entre les deux images.

Collins et *al* [18] proposent la différence temporelle entre trois images successives. Soient $I(x,y,t)$ la valeur de l'intensité du pixel (x,y) à l'instant t et $\tau(x,y,t)$ la valeur du seuil assigné au pixel (x,y) à l'instant t qui permet de détecter un changement significatif. Un pixel est classé en mouvement si :

$$| I(x,y,t) - I(x,y,t-1) | \geq \tau(x,y,t) \quad \text{et} \quad | I(x,y,t) - I(x,y,t-2) | \geq \tau(x,y,t)$$

Cette méthode ne permet pas de détecter le mouvement dans les zones uniformes intérieures à l'objet (Figure 1.4). En plus, elle ne fonctionne pas dans plusieurs cas, pour différentes raisons, telles que: la présence de bruit du capteur et les changements de luminosité de la scène qui modifient les intensités des pixels.

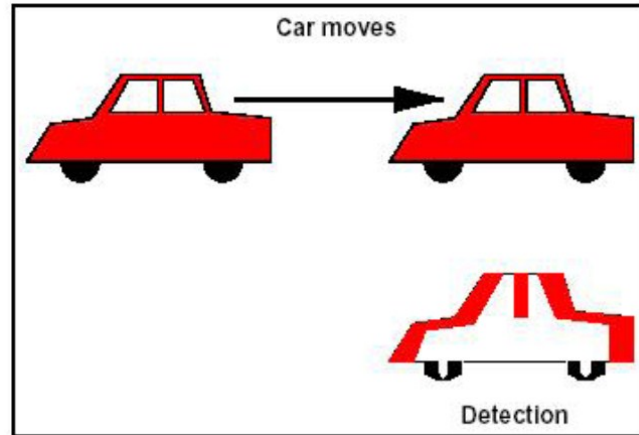


Figure 1.4 : Détection de mouvement par différence temporelle [18].

2-2-2- Le flux optique

Le flux optique est un champ de vecteur des vitesses apparentes des objets de la scène sur le plan image (Figure 1.5). Il permet de déterminer le mouvement de chaque pixel appartenant à une image de la séquence d'images.

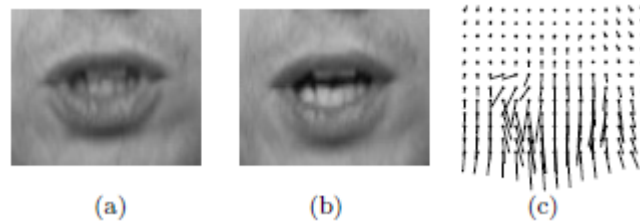


Figure 1.5 : Le flux optique [29]. (a et b) Région de la bouche sur deux images consécutives. (c) Champ de vecteur de vitesses.

L'estimation du flux optique est basée sur l'hypothèse de l'intensité constante c-à-d que l'intensité observée d'un objet ne change pas lorsque celui-ci est en mouvement. Cette hypothèse est valable pour les objets à modèle de réflectance lambertienne avec un faible mouvement. Cette méthode suppose donc, que le mouvement est instantané et faible d'une image à l'autre et que les changements observés entre deux images sont uniquement causés par le mouvement.

Le calcul du flux optique est basé essentiellement sur un calcul du gradient. Selon l'hypothèse de l'intensité constante [12], on peut écrire l'intensité d'un pixel X au temps t :

$$I(X, t) = I(X + \Delta X, t + \Delta t)$$

où ΔX est le déplacement effectué dans un intervalle de temps Δt

Le développement en série de Taylor donne :

$$I(X, t) = I(X, t) + \nabla I \cdot \Delta X + \Delta t I_t + O^2$$

où $\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$, $I_t = \frac{\partial I}{\partial t}$ sont les dérivées partielles d'ordre 1 de la fonction

intensité respectivement par rapport à (x, y) et t et O^2 la dérivée d'ordre 2 qui est supposée être négligeable.

En soustrayant $I(X, t)$ des deux côtés de l'équation du développement de Taylor et en divisant par Δt , on obtient :

$$\nabla I^T v + I_t = 0 \quad \text{avec} \quad v = \left(\frac{\Delta x}{\Delta t}, \frac{\Delta y}{\Delta t} \right) \quad \text{le vecteur vitesse.}$$

C'est une équation qui s'apparente à l'équation d'une droite dans l'espace 2D. Elle donne seulement la composante de la vitesse dans la direction du gradient spatial, il s'agit de la vitesse normale v_n . Le flux normal est donc fonction du gradient spatio-temporel de l'image :

$$v_n = \frac{-I_t}{\|\nabla I\|}$$

Cette caractéristique induit une sensibilité du flux optique par rapport au bruit dans l'image en plus du temps machine que la complexité de son calcul induit. De ce fait, il est peu utilisé en vidéosurveillance [18].

2-3- Les méthodes hybrides

Les méthodes hybrides de détection de mouvement sont une combinaison des méthodes basées sur la modélisation de l'arrière-plan et les méthodes basées sur des mesures des zones de mouvement.

Collins et al [18] proposent dans la conception du système VSAM de combiner la méthode différentielle utilisant trois images et une méthode adaptative de soustraction d'arrière-plan. La première permet de déterminer les régions où il y a un mouvement et la seconde permet d'extraire la totalité de la région en mouvement en comblant les trous.

La différence temporelle entre trois images successives (§ 2.2.1) classe un pixel $I(x,y,t)$ en mouvement si :

$$| I(x,y,t) - I(x,y,t-1) | \geq \tau(x,y,t) \quad \text{et} \quad | I(x,y,t) - I(x,y,t-2) | \geq \tau(x,y,t)$$

Le problème est que les pixels appartenant à l'intérieur de l'objet en mouvement ayant une intensité uniforme ne sont pas extraits parmi les pixels en mouvement. Ils sont regroupés dans une boîte englobante R . Ainsi les pixels intérieurs n'ayant pas été détectés comme pixels en mouvement peuvent être classés correctement en appliquant une méthode adaptative de soustraction d'arrière-plan. Soit $B(x,y,t)$ la valeur de l'intensité du pixel (x,y) dans l'image arrière-plan au temps t . On constitue un blob b_t contenant les pixels de R ayant une différence significative avec l'arrière-plan au temps t . Soit :

$$b_t = \{ (x,y) \in R \mid |I(x,y,t) - B(x,y,t)| > \tau(x,y,t) \}$$

L'image arrière-plan est initialisée avec la première image de la séquence $I(x,y,0)$ et est mise-à-jour par :

$$B(x,y,t+1) = \begin{cases} \alpha B(x,y,t) + (1-\alpha)I(x,y,t) & \text{si } (x,y) \text{ statique} \\ B(x,y,t) & \text{si } (x,y) \text{ en mouvement} \end{cases}$$

Le seuil τ est initialisé par une valeur non nulle, il est mis-à-jour par :

$$\tau(x,y,t+1) = \begin{cases} \alpha \tau(x,y,t) + (1-\alpha)(5|I(x,y,t) - B(x,y,t)|) & \text{si } (x,y) \text{ statique} \\ \tau(x,y,t) & \text{si } (x,y) \text{ en mouvement} \end{cases}$$

Si la position de chaque pixel qui n'est pas en mouvement est considérée comme une série temporelle, alors $B(x,y,t)$ est analogue à une moyenne temporelle locale des valeurs d'intensité et $\tau(x,y,t)$ est analogue à 5 fois (valeur empirique) la déviation temporelle locale des valeurs d'intensités.

Cette méthode hybride combinant la méthode différentielle et la méthode adaptative de soustraction d'arrière-plan donne de bons résultats (Figure 1.6).



Figure 1.6: Détection de mouvement par méthode hybride [18]

(a) Image originale, (b) Régions en mouvement détectées.

2-4-Élimination des ombres

La segmentation des objets mobiles permet de les extraire de l'arrière-plan statique. Cette opération extrait les objets accompagnés de leurs ombres, ce qui constitue une difficulté majeure pour les processus de suivi et de reconnaissance. Plusieurs méthodes de segmentation des mouvements permettent de détecter et éliminer les ombres.

2-4-1- Seuillage en HSV [19]

C'est une méthode déterministe sans modèle d'image en référence. Elle suppose que l'ombre cause des changements faibles dans les canaux H (la teinte) et S (la saturation), mais plus fort dans le canal V (l'intensité). Donc moyennant un seuillage sur ces valeurs, si un pixel présente un changement très grand, il ne s'agit pas d'une ombre.

2-4-2- Méthode statistique [22]

Cette méthode considère que l'ombre affecte les composantes RGB du pixel d'une manière proportionnelle avec une quasi-préservation de la couleur dominante produisant ainsi un effet de semi-transparence. Elle utilise deux mesures, soient la distorsion de l'intensité et la distorsion chromatique.

Soient $I(x) = [I_r(x), I_g(x), I_b(x)]$ les composantes RGB d'un pixel x de l'image courante I et $B(x) = [B_r(x), B_g(x), B_b(x)]$ les composantes RGB du pixel x sur l'image de référence B .

La distorsion de l'intensité est définie par : $\delta Br(x) = \|B(x)\| - \frac{I(x) \bullet B(x)}{\|B(x)\|}$

La distorsion chromatique est définie par : $\delta Cr(x) = \arccos \left[\frac{I(x) \bullet B(x)}{\|I(x)\| \|B(x)\|} \right]$

où \bullet représente le produit scalaire.

Un pixel est classé comme appartenant à une ombre si :

$$\delta Br(x) < (1 - \tau_1) \|B(x)\| \quad \text{et} \quad \delta Cr(x) < \theta$$

où τ_1 et θ sont des seuils déterminés d'une façon empirique.

La figure (1.7) illustre le principe de cette méthode :

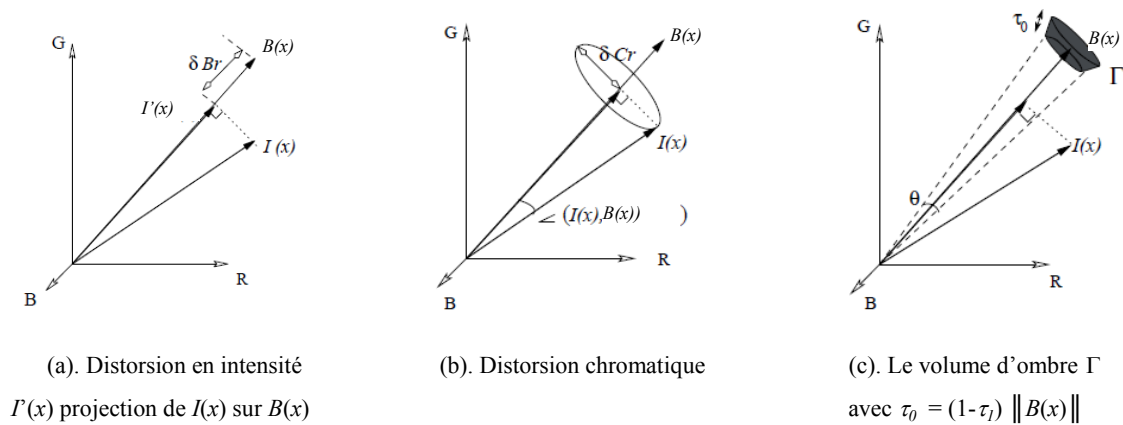


Figure 1.7: Calcul des distorsions et volume d'ombre (adaptée de [22]).

2-4-3- Méthode du gradient et chromaticité

La présence de l'ombre cause un changement dans l'intensité, mais pas dans la chromaticité. La méthode proposée par McKenna et *al.* [44], effectue une soustraction de l'arrière-plan avec mise à jour en utilisant la chromaticité plutôt que les intensités de couleurs RGB.

Pour un pixel (x,y) , on calcule pour chaque canal RGB :

- le gradient $(r_x, r_y), (g_x, g_y), (b_x, b_y)$ par une convolution avec le masque de Sobel [59],
- la moyenne du gradient $(\mu_{xr}, \mu_{yr}), (\mu_{xg}, \mu_{yg}), (\mu_{xb}, \mu_{yb})$,
- la variance de son module $\sigma_{gr}^2, \sigma_{gb}^2, \sigma_{gg}^2$ ainsi que les moyennes $\bar{\sigma}_{gr}^2, \bar{\sigma}_{gb}^2, \bar{\sigma}_{gg}^2$ sur toute l'image.

Un pixel est classé avant-plan si une des relations suivantes est vraie :

$$\begin{aligned} \sqrt{(r_x - \mu_{xr})^2 + (r_y - \mu_{yr})^2} &> 3 \max \{ \sigma_{gr}, \bar{\sigma}_{gr} \} \quad \text{ou} \\ \sqrt{(r_x - \mu_{xg})^2 + (r_y - \mu_{yg})^2} &> 3 \max \{ \sigma_{gg}, \bar{\sigma}_{gg} \} \quad \text{ou} \\ \sqrt{(r_x - \mu_{xb})^2 + (r_y - \mu_{yb})^2} &> 3 \max \{ \sigma_{gb}, \bar{\sigma}_{gb} \} \end{aligned}$$

3- Le suivi des objets

Le suivi des objets est une étape importante dans un système de vidéosurveillance. Il consiste à faire une estimation de la position des objets mobiles dans le temps et l'espace du champ de vision afin de déterminer la trajectoire des objets mobiles à travers les images successives d'une séquence. Cela se fait par la mise en correspondance d'un objet sur l'image avec le même objet sur l'image successive. Pour cela, le suivi doit résoudre des problèmes tels que l'apparition de nouveaux objets mobiles dans la scène ou leur disparition, l'occultation partielle ou totale, la fusion et la séparation des blobs.

Le suivi peut se faire sur une séquence d'images prise par une caméra fixe ou une caméra en mouvement ou en multicaméra. Il peut aussi se faire avec soustraction de l'arrière-plan ou sans soustraction de l'arrière-plan. Plusieurs travaux ont été proposés dans le domaine du suivi :

3-1-Méthode de recouvrement

Fuentes et *al* [27] supposent une caméra fixe et une vitesse de déplacement des objets relativement faible par rapport à la fréquence de capture des caméras. Ils procèdent à la soustraction de l'arrière plan et la détermination des composantes connectées, ils superposent deux images consécutives, ainsi des parties d'un même objet à suivre seront en recouvrement.

3-2- Utilisation du filtre de Kalman

Forsyth et *al* [26] utilisent le Filtre de Kalman pour le suivi, sans effectuer de soustraction d'arrière plan. Le filtre de Kalman [37] est un estimateur récursif linéaire gaussien de l'état d'un système. L'état courant d'un système est estimé seulement à partir de l'état précédent et des mesures actuelles, l'historique des observations et des estimations n'est pas requis.

Dans l'application du filtre de Kalman au processus de suivi, on suppose que le déplacement d'un objet d'une image à l'autre se fait suivant un modèle connu, et que les observations de l'objet correspondent à ce modèle auquel s'ajoute un bruit gaussien. Le filtre de Kalman donne alors une prédiction de la prochaine position de l'objet en déplacement. Il se déroule de manière cyclique et possède deux phases distinctes: la prédiction et la correction (Figure 1.8). La phase de prédiction consiste à produire une estimation de l'état courant en utilisant l'état précédent. Lors de la phase de correction, les observations de l'instant courant sont utilisées pour corriger l'état prédit dans le but d'obtenir une estimation plus précise.

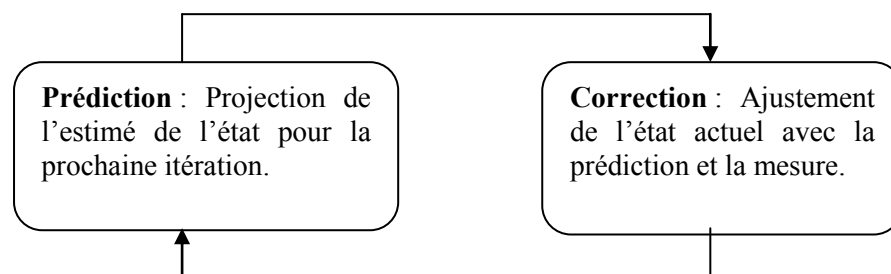


Figure 1.8 : Le cycle du filtre de Kalman [52].

Le Filtre de Kalman est efficace pour le suivi du trafic routier, il est optimal pour un modèle linéaire et un bruit gaussien, mais peu efficace en cas de changements brusques de la direction de la cible suivie et en cas d'occlusions.

3-3- Utilisation du filtre à particules

Isard et *al* [36] utilisent le filtre à particules pour le suivi de trajectoires avec des changements brusques et où les cibles peuvent être multiples. Le filtre à particule permet de lever la contrainte de linéarité et du modèle gaussien qu'impose le filtre de Kalman.

Dans cette méthode, on choisit initialement une région modèle et on la modélise selon son histogramme d'intensité. On choisit un nombre de particules qu'on positionne aléatoirement sur l'image suivante à partir de la région modèle. Pour le suivi, on compare les particules avec la région modèle moyennant une distance qui permet de déterminer la similarité entre les particules et le modèle. On retient les particules qui représentent un taux de similarité élevé et la moyenne pondérée des particules ayant la plus grande similarité devient la région à suivre pour les prochaines trames et le processus se répète.

Une revue exhaustive des méthodes de suivi a été publiée par Yilmaz et *al* [63].

4- La reconnaissance

Cette dernière phase d'un système de vidéosurveillance concerne, selon le domaine de l'application, la reconnaissance des activités, la reconnaissance des comportements, l'identification,... Il s'agit d'une opération de classification supervisée ou non supervisée. Plusieurs outils mathématiques sont utilisés dans la reconnaissance :

Les modèles de Markov cachés sont des automates à état stochastique adaptés à la classification des séquences d'actions. Ils ont été utilisés pour la reconnaissance de la parole [49], l'analyse d'activités humaines [31].

Par exemple, Fuentes et al [27] proposent d'abord de soustraire l'arrière plan en se basant sur le contraste de luminance de l'objet et celle de l'arrière-plan. L'identification des activités se base alors sur la trajectoire des blobs. Ils supposent que la scène contient peu de personnes en mouvement donc moins de problèmes d'occultation. La trajectoire de chaque blob est déterminée par l'intersection des boîtes englobantes sur deux images consécutives en tenant compte des séparations/fusions des blobs à travers deux images successives. Les événements seront détectés selon une combinaison d'évènements élémentaires sur les images, par exemple : à l'évènement "Un bagage est abandonné" correspondent les règles suivantes : un blob se sépare en deux (personne et bagage) pendant une durée définie, puis un des blobs est stationnaire (le bagage), le blob correspondant à la personne est en mouvement tout en s'éloignant du blob stationnaire.

D'autres travaux spécifiques à une seule activité ont aussi été menés par de nombreux chercheurs :

4-1- Prise de médicaments [35]

Cette méthode est composée de trois opérations majeures : la détection et le suivi de la bouche, la détection et le suivi des mains et la détection et le suivi des flacons de médicaments. Après soustraction de l'arrière-plan et élimination des ombres, cette méthode segmente la couleur peau, localise le visage par détection des yeux avec des descripteurs (ondelettes) de Harr [61], puis détecte la bouche (couleur plus rouge) et en fait le suivi par un filtre de Kalman. Les régions des mains sont détectées grâce à la présence des doigts qui lui donnent une intensité de contours importante par rapport aux bras. L'auteur utilise l'opérateur de Canny [16] pour détecter ces contours.

Les flacons de médicaments utilisés sont de couleur uniforme et segmentés par seuillage. La correspondance des flacons dans deux images consécutives est faite par la distance minimale au vecteur de couleur moyenne.

La reconnaissance de prise de médicaments se fait par reconnaissance d'évènements composés d'activités élémentaires telles que : A_i : une main touche le flacon i , B_i : deux mains touchent le flacon i , C : une main approche le visage, D : une main touche la région de la

bouche...La prise de médicaments peut être détectée par la séquence A_iB_iCD en tenant compte de la durée des événements et des activités.

4-2- Reconnaissance de démarches [54]

Cette méthode s'applique seulement à des vues latérales et se base sur le fait que le nombre de pixels de la région du marcheur varie en fonction de sa pose. Il est plus grand lorsque les jambes sont séparées que lorsqu'elles sont en recouvrement. Cette méthode procède d'abord à la soustraction de l'arrière plan, la région binaire de l'être humain obtenue est mise à l'échelle (hauteur = 128 pixels). Le calcul se fait sur la moitié inférieure de la région soit la région des jambes, on y calcule le nombre de pixels correspondant à l'avant-plan. La figure 1.9 montre le nombre de pixels de l'avant plan au cours de la séquence d'images.

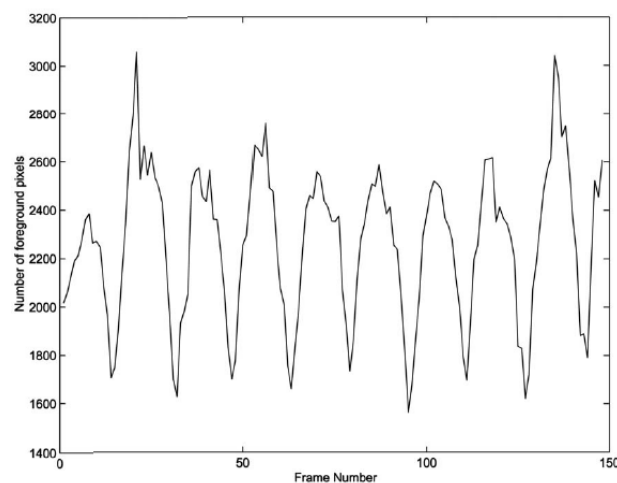


Figure 1.9: Nombre de pixels des jambes en mouvement [54].

La figure 1.9 montre une variation périodique du nombre de pixels appartenant à la silhouette des jambes en fonction de leur pose (recouvrement / séparation). La période correspondant à deux foulées consécutives est évaluée par la moyenne des deux médianes de la distance entre 2 minimas consécutifs en ignorant le premier minima pour le calcul de la première médiane. Les démarches sont comparées sur cette période en comparant le ratio de l'intersection sur l'union des pixels de la région du marcheur.

5- La vidéosurveillance appliquée à la détection des chutes

5-1- Intérêts et objectifs

La population du Québec (et de plusieurs pays) connaît un vieillissement marqué de sa population qui se traduira par une hausse importante du nombre et de la proportion des personnes âgées de 65 ans et plus. Ce phénomène de vieillissement est la résultante de la baisse marquée du nombre de naissances depuis le milieu des années soixante et d'une augmentation soutenue de l'espérance de vie à la naissance. D'après les données de l'Institut de Santé publique du Québec, la population de 65 ans et plus a atteint 1 253 837, en 2011 et ce nombre est en hausse, il atteindrait en 2026, les 2 028 994 (Tableau I) [5].

Tableau I : Nombre et importance relative des aînés, selon l'âge, Québec, 1981 à 2041 [5].

Âge des babyboomers	Effectif						Importance relative en (%)		
	Année	65-74 ans	75-84 ans	85 ans et plus	65 ans et plus	Tous âges	65 ans et plus	75 ans et plus	85 ans et plus
13-36 ans	1981	372 058	163 564	37 587	573 209	6 547 705	8,8	3,1	0,6
18-41 ans	1986	411 705	198 193	47 907	657 805	6 708 468	9,8	3,7	0,7
23-46 ans	1991	479 772	239 159	63 341	782 272	7 064 586	11,1	4,3	0,9
28-51 ans	1996	523 956	267 677	77 329	868 962	7 246 896	12,0	4,8	1,1
33-56 ans	2001	549 395	320 421	95 295	965 111	7 396 988	13,0	5,6	1,3
38-61 ans	2006	575 621	370 543	118 917	1 065 081	7 631 552	14,0	6,4	1,6
43-66 ans	2011	691 887	403 368	158 582	1 253 837	7 946 837	15,8	7,1	2,0
48-71 ans	2016	851 901	441 567	194 491	1 487 959	8 227 004	18,1	7,7	2,4
53-76 ans	2021	980 166	545 548	222 220	1 747 934	8 470 571	20,6	9,1	2,6
58-81 ans	2026	1 091 008	682 435	255 551	2 028 994	8 678 345	23,4	10,8	2,9
63-86 ans	2031	1 137 338	797 724	327 577	2 262 639	8 838 257	25,6	12,7	3,7
68-91 ans	2036	1 041 081	901 788	416 256	2 359 125	8 956 967	26,3	14,7	4,6
73-96 ans	2041	954 078	950 034	505 944	2 410 056	9 049 191	26,6	16,1	5,6

Ces prédictions incitent la communauté scientifique à développer des solutions qui permettent le maintien des aînés à domicile le plus longtemps possible, afin de pouvoir bénéficier d'une meilleure qualité de vie. Ainsi, la maison intelligente suscite de plus en plus l'attention des chercheurs.

Une maison intelligente consiste en quatre fonctions [64] : la première est de soutenir les personnes handicapées selon leur cas, par un système de contrôle de voix, des balises

audibles, des écrans tactiles. La deuxième fonction consiste en l'utilisation des systèmes de monitoring pour la surveillance des signes physiologiques (pression artérielle, température,...). Troisièmement, la supervision de la prise de médicament et de nourriture. Enfin, la maison intelligente doit assurer un confort à ses occupants, à travers un environnement contrôlé (chauffage, lumière, air conditionné,...).

Dans ce volet, des travaux ont été menés en vidéosurveillance afin de détecter les chutes chez les personnes âgées et générer une alerte afin de secourir de telles personnes vivant seules. La détection de chutes présente donc un intérêt pour le maintien à domicile des aînés. Le but de ces recherches est de détecter une chute parmi des mouvements ordinaires de la vie quotidienne, comme s'asseoir, s'accroupir, se pencher,... Selon les auteurs, une chute présente des caractéristiques particulières qui permettent de la distinguer de tout autre mouvement et contribuent ainsi au processus de sa détection.

5-2- Caractéristiques d'une chute

L'évènement chute est caractérisé par quatre phases distinctes [52][47] (Figure 1.10):

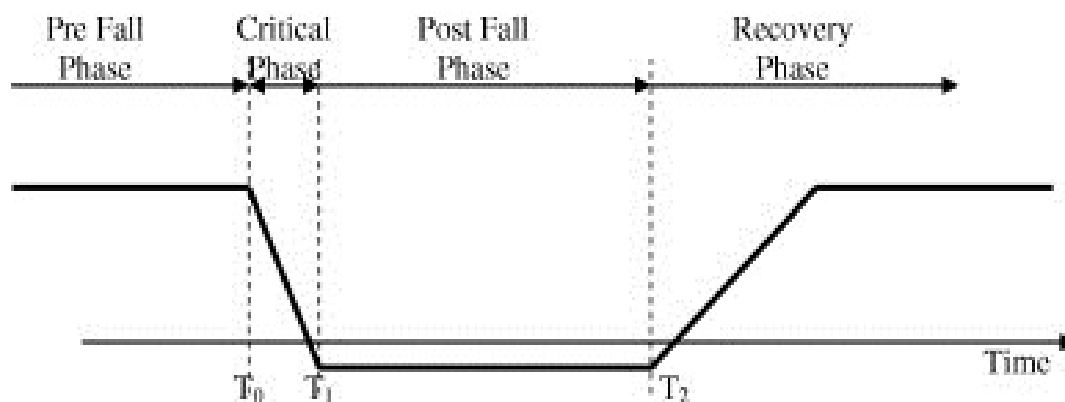


Figure 1.10 : Les quatre phases d'une chute [47].

Phase de pré-chute : La personne se déplace avec une vitesse peu variable, dans son environnement afin d'exercer ses activités quotidiennes. Ces activités peuvent être interférées par des mouvements soudains comme s'asseoir, s'abaisser pour ramasser un objet. Ce type de mouvement ne devrait pas être détecté comme une chute.

Phase critique : Cette phase est caractérisée par sa courte durée; elle correspond à un mouvement soudain avec une vitesse considérable et en direction du sol.

Phase de post-chute : Après sa chute, la personne reste généralement immobile sur le sol.

Phase de rétablissement : Il arrive que la personne se relève d'elle-même de sa chute ou grâce à l'aide d'une autre personne.

5-3- Méthodes de détection de chutes par vidéosurveillance

Anderson et *al* [8] ont développé une méthode qui détecte les chutes en comparant les dimensions de la boîte englobante de la silhouette de la personne par rapport à l'ensemble de l'image. Cette méthode ne résout pas le problème de l'occultation, si une personne fait une chute derrière un meuble par exemple.

Lee et Mihailidis [38] définissent d'abord des zones d'inactivités habituelles dans la pièce telles que le lit, le canapé..., puis analysent la silhouette de la personne et sa vitesse en utilisant des seuils caractéristiques selon que la zone soit d'activité ou d'inactivité.

Rougier et *al* [50] ont développé une méthode de détection des chutes basée sur la déformation de la silhouette lors d'une chute. Il s'agit de mettre en correspondance des parties similaires de la silhouette d'une personne entre deux images consécutives en modélisant la distribution des données d'activités normales d'une personne par un mélange de gaussiennes.

Rougier et *al* [51] ont aussi développé une méthode de détection des chutes qui considère la tête de la personne comme étant une ellipsoïde 3D qui est projetée dans le plan

image à l'aide des paramètres internes de la caméra. La tête est suivie à l'aide d'un filtre à particules. La caméra étant calibrée, grâce à la connaissance de sa position par rapport au sol, un objet (la tête de la personne) est localisé par rapport au sol.

Auvinet et *al* [11] ont développé une méthode de détection des chutes qui utilise jusqu'à huit caméras positionnées autour d'une pièce où se déroulent les expériences de chutes. Ils procèdent en trois étapes, la première consiste à calibrer les caméras afin de déterminer leurs paramètres intrinsèques et extrinsèques. La deuxième consiste à soustraire l'arrière plan au niveau de chaque caméra puis faire une fusion des différents résultats pour reconstruire la silhouette en 3D. Troisièmement, la détection des chutes se fait selon un indicateur calculé en comparant le volume des 40 premiers centimètres de la silhouette par rapport au volume total depuis le sol. Une valeur élevée de cet indicateur génère une alerte de chute.

5-4- Méthodes de détection de chutes sans vidéosurveillance

- **Capteurs portés par la personne** [7] [52]: La personne porte un capteur ou un bouton poussoir sous forme de bracelet par exemple. En cas de chute, la personne active ce bouton pour générer une alerte et appeler au secours. Cette méthode est efficace seulement si la personne est consciente et est en mesure d'actionner le bouton.

- **Capteurs au sol** [57][65]: Des capteurs de vibrations sont fixés au sol afin de détecter des chutes. Cette méthode dépend de la dynamique du sol et reste difficile à mettre en œuvre.

6- Conclusion

Dans cette étude bibliographique, nous nous sommes intéressés à la vidéosurveillance. Nous y avons exposé les différentes étapes du processus de vidéosurveillance, depuis la détection du mouvement sur les images, jusqu'à la reconnaissance d'évènements, en passant par la phase cruciale du suivi des objets mobiles et ce en nous basant sur des travaux de différents chercheurs. Nous avons aussi présenté quelques travaux liés à la détection d'évènement particulier tel que la chute chez les personnes âgées. Beaucoup de problèmes restent encore non résolus, notamment au niveau de la détection de mouvement lorsque *la caméra est en mouvement*.

Dans le chapitre suivant, nous présentons les fondements théoriques de notre algorithme de détection de chutes, à partir d'images prises par une caméra en rotation.

Cette revue des différentes méthodes n'est pas exhaustive, d'autres travaux sont dans [15][48][63].

Chapitre 2

Aspects Théoriques de l'Algorithme de Détection des Chutes

1- Introduction

La détection des chutes a été modélisée selon plusieurs concepts, qui pour la plupart se basent sur la soustraction de l'arrière-plan pour la détection du mouvement dans la scène. Ces méthodes s'avèrent inefficaces si la caméra est en mouvement, car cela engendre un changement de l'arrière-plan. Afin de pallier à cet inconvénient, nous proposons dans ce chapitre, une méthode qui s'affranchit de la soustraction de l'arrière-plan dans le processus de détection de mouvement, afin de pouvoir réaliser de la vidéo surveillance avec une caméra en rotation.

En effet, nous travaillons sur des images d'une séquence vidéo contenant du mouvement et prises par une caméra en rotation pure, où nous procédons par extraction des points caractéristiques sur deux images consécutives. Après leur mise en correspondance, ces points sont classés en mouvement s'ils ne correspondent pas à l'homographie (qui sera présentée à la section 7, p. 46) reliant les deux images consécutives. La détection de chute se fait par évaluation de la vitesse du mouvement.

2- Problématique et objectif

Dans ce travail, nous cherchons à détecter les chutes dans une séquence d'images vidéo représentant une scène intérieure. L'application étant dédiée à des personnes âgées, nous émettons les deux hypothèses suivantes:

- 1- La scène présente une seule personne en mouvement, car si cette personne se trouve en compagnie d'une autre personne, celle-ci pourrait donner l'alerte en cas de chute.
- 2- Tout mouvement brusque est considéré comme une chute quelque soit sa direction, car les mouvements ordinaires chez une personne âgée sont lents et ne peuvent être brusques.

Le modèle sur lequel nous travaillons est constitué d'une caméra fixée en haut du mur et disposée sur un support plan incliné vers le bas d'un angle α par rapport au plan horizontal (Figure 2.1). La caméra effectue des rotations pures d'un angle θ , sur ce plan incliné (Figure 2.2), afin de capter des images de la scène en mouvement avec des points de vue différents.

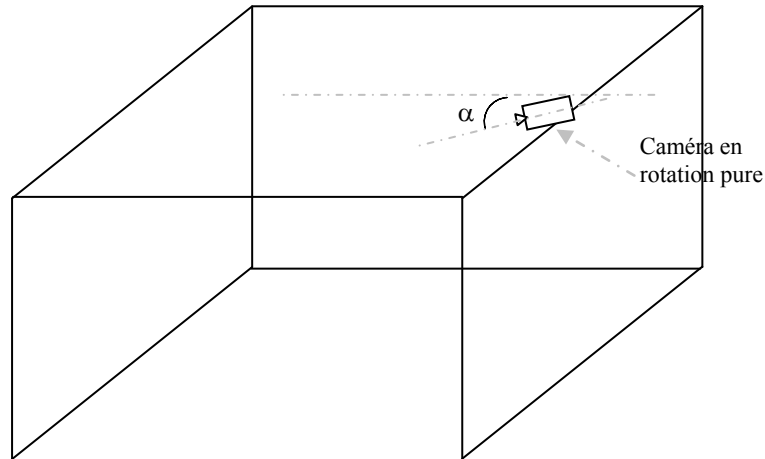


Figure 2.1 : Disposition de la caméra dans la pièce.

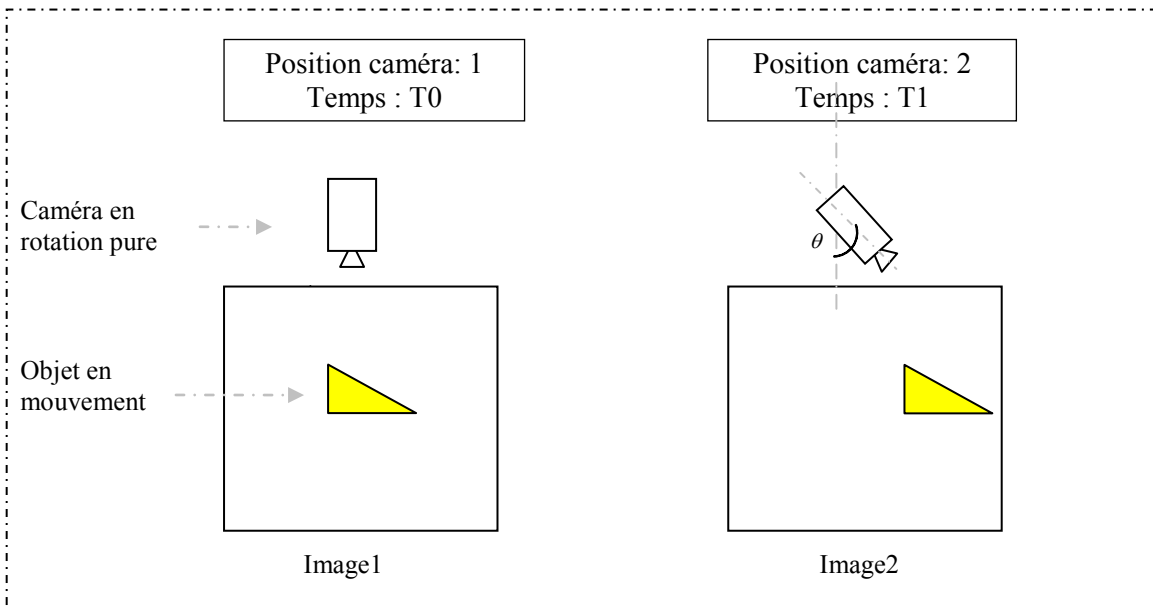


Figure 2.2 : Objet en mouvement sur deux images prises par une caméra en rotation pure.

L'objectif fixé est de développer une méthode de détection de mouvement de chutes à partir d'images successives prises par la caméra en rotation. Cet algorithme s'inspire des travaux de Xu et *al* [62] qui présentent la segmentation du mouvement par extraction de l'arrière-plan, en exploitant les positions pan et tilt de la caméra fournies par le moteur la faisant tourner.

Notre algorithme est basé sur une étude théorique qui ne nécessite pas le calibrage de la caméra, ni la connaissance précise de l'angle de rotation θ . Un intervalle de valeurs de cet angle est défini par expérimentation.

Pour atteindre notre objectif, nous procédons selon la stratégie suivante :

- À partir d'une étude théorique basée essentiellement sur le calcul homographique (section 7, p.46), développer l'algorithme de détection de mouvement et reconnaissance de chutes.
- Expérimenter l'algorithme sur des images générées par ordinateur.
- Tester la robustesse de l'algorithme sur des images réelles.

3- Intérêts du mouvement de rotation de la caméra

La caméra effectue une rotation pure d'angle θ et dont le centre correspond au centre optique de la caméra O_c . Les mouvements de rotation pure qu'effectue la caméra lui permettent de faire l'acquisition d'une séquence d'images reliées deux à deux par une homographie. Une estimation de cette homographie par une méthode robuste telle que RANSAC (Random Sample Consensus) [25] permet de retenir les points immobiles sur l'image et ainsi extraire les points en mouvement dans la scène.

4- Modélisation de la caméra [33][59]

Dans cette modélisation, nous définissons 3 systèmes de coordonnées distincts (Figure 2.3) :

- Un repère 3D $(O_w X_w Y_w Z_w)$ orthonormé lié à la scène (au monde): c'est un système de référence absolu défini par l'utilisateur et tous les objets du monde y compris la caméra, sont définis par rapport à ce système de coordonnées. L'indice w se rapporte au monde (*world*).
- Un repère 3D $(O_c X_c Y_c Z_c)$ orthonormé dont l'origine est le centre de la caméra, l'indice c se rapporte à la caméra.
- Un repère 2D $(O_s X_s Y_s)$ orthonormé dont l'origine est en haut à gauche du plan image et dont les axes X_s et Y_s sont parallèles respectivement aux lignes et colonnes de l'image. L'indice s se rapporte à l'image (*screen*).

La caméra est modélisée en position canonique (Figure 2.3) où l'axe Z_c est vers l'avant de la caméra, l'axe Y_c est vers le haut et l'axe X_c est vers la gauche.

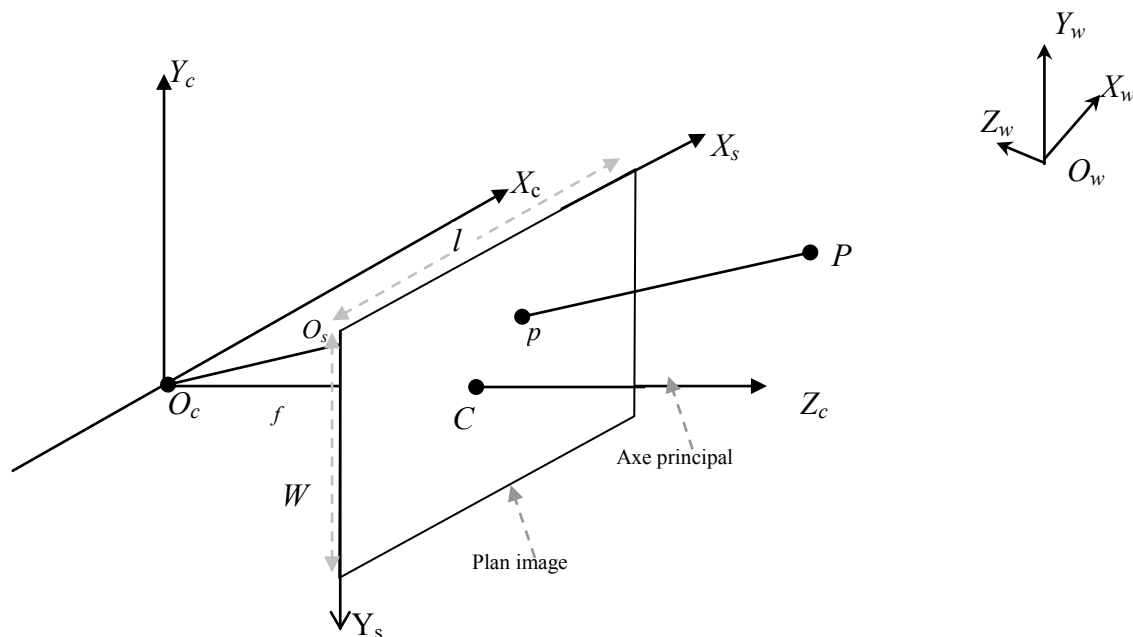


Figure 2.3 : Modélisation de la caméra et projection d'un point 3D.

où O_c : Centre de la caméra (position de la caméra dans le repère $(O_w X_w Y_w Z_w)$.
 $O_c Z_c$: axe optique de la caméra; c'est l'axe principal.
 C : Point principal (intersection entre l'axe optique et le plan image).
 P : Un point de la scène 3D.
 p : Projection du point P dans l'image.
 f : Distance focale entre le plan image et le centre O_c de la caméra ($f = O_c C$).
 W : Largeur du plan image.
 l : Longueur du plan image.

La largeur du plan image W et la focale f de la caméra, permettent de déterminer l'angle de vue β de la caméra (Figure 2.4).

$$\tan\left(\frac{\beta}{2}\right) = \frac{W}{2f} \quad (2.1)$$

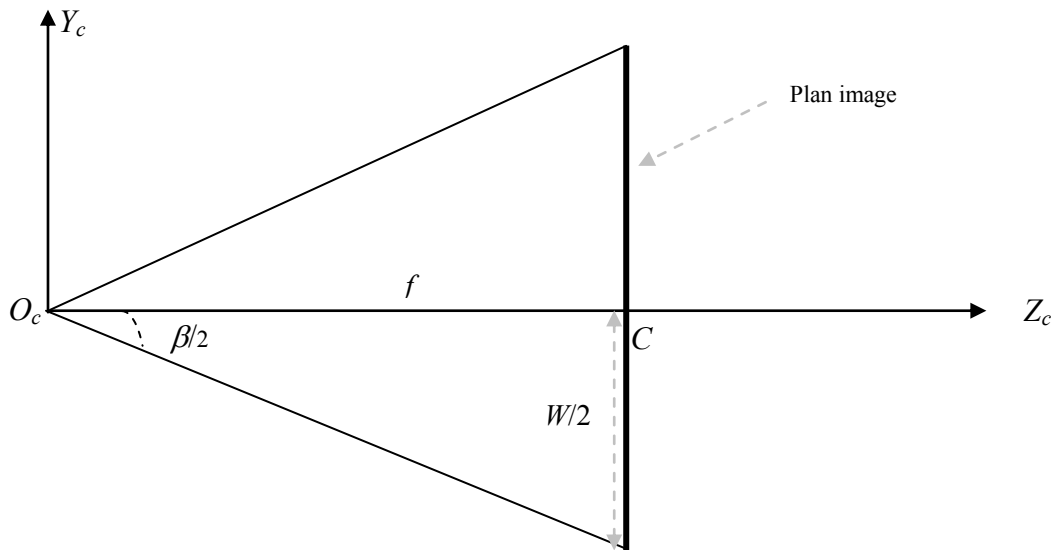


Figure 2.4. : Angle de vue de la caméra (adaptée de [59]).

4-1- Projection d'un point 3D sur l'image [33][59]

La projection perspective d'un point 3D $P=(x_w, y_w, z_w)$ sur l'image, se fait en lui associant un point 2D $p = (x, y)$ de l'image. Elle se fait selon deux transferts en tenant compte d'abord des paramètres externes et puis des paramètres internes de la caméra. Ainsi une matrice de transformation \mathbf{M} appelée matrice de pose de la caméra et déterminée par les paramètres externes de la caméra (position de la caméra et son orientation par rapport au repère du monde), fait le transfert du repère du monde vers le repère de la caméra. La position de la caméra est modélisée par le vecteur translation \mathbf{t} du centre optique de la caméra par rapport au repère du monde et l'orientation de la caméra est modélisée par sa matrice de rotation \mathbf{R} par rapport au repère du monde. Dans ce premier transfert, le point P du monde est équivalent au point P_c dans le repère de la caméra. Le deuxième transfert se fait du repère caméra vers celui de l'image, en tenant compte des paramètres internes de la caméra qui sont la distance focale f et les coordonnées du point principal C . Ce transfert est effectué par une matrice \mathbf{K} appelée matrice de calibrage de la caméra. La figure 2.5 illustre ces deux transferts.

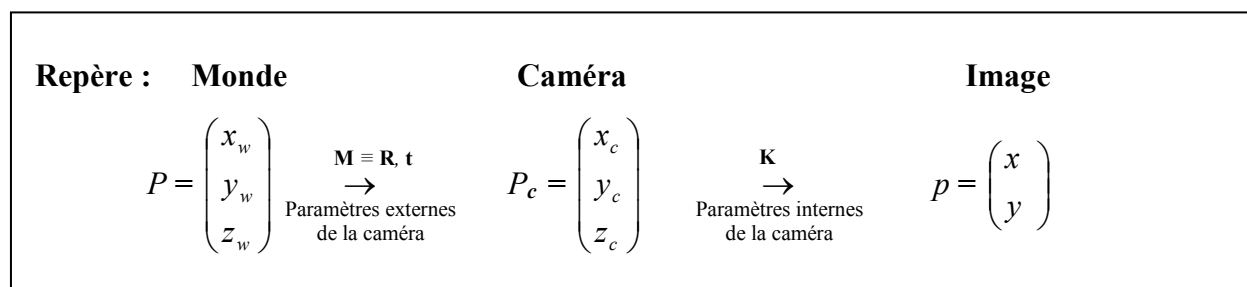


Figure 2.5 : Projection perspective d'un point du monde par changement de repère.

Calcul de la matrice \mathbf{M}

La caméra subit par rapport au repère du monde, une translation $\mathbf{t} = (t_x, t_y, t_z)$ et une rotation $\mathbf{R} = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix}$ résultant de la combinaison des 3 rotations selon les 3 axes du repère du monde, définies par les angles d'Euler:

$$\mathbf{R}_x(\gamma_1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma_1) & -\sin(\gamma_1) \\ 0 & \sin(\gamma_1) & \cos(\gamma_1) \end{pmatrix} \quad ; \quad \mathbf{R}_y(\gamma_2) = \begin{pmatrix} \cos(\gamma_2) & 0 & \sin(\gamma_2) \\ 0 & 1 & 0 \\ -\sin(\gamma_2) & 0 & \cos(\gamma_2) \end{pmatrix}$$

$$\mathbf{R}_z(\gamma_3) = \begin{pmatrix} \cos(\gamma_3) & -\sin(\gamma_3) & 0 \\ \sin(\gamma_3) & \cos(\gamma_3) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Ainsi tout point P du monde se transfère dans le repère de la caméra en fonction de ces deux transformations \mathbf{R} et \mathbf{t} selon l'équation (2.2).

$$P = \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} \rightarrow P_c = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} \begin{pmatrix} x_w - t_x \\ y_w - t_y \\ z_w - t_z \end{pmatrix} \quad (2.2)$$

Le développement du produit donne :

$$P = \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} \rightarrow P_c = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} r_1 x_w + r_2 y_w + r_3 z_w - r_1 t_x - r_2 t_y - r_3 t_z \\ r_4 x_w + r_5 y_w + r_6 z_w - r_4 t_x - r_5 t_y - r_6 t_z \\ r_7 x_w + r_8 y_w + r_9 z_w - r_7 t_x - r_8 t_y - r_9 t_z \end{pmatrix} \quad (2.3)$$

Dans l'espace projectif qui sera présenté à la section 7, s'ajoute à chaque point une coordonnée homogène égale à 1, d'où le changement de la notation des points:

$$\tilde{P} = \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \rightarrow \tilde{P}_c = \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} r_1 x_w + r_2 y_w + r_3 z_w - r_1 t_x - r_2 t_y - r_3 t_z \\ r_4 x_w + r_5 y_w + r_6 z_w - r_4 t_x - r_5 t_y - r_6 t_z \\ r_7 x_w + r_8 y_w + r_9 z_w - r_7 t_x - r_8 t_y - r_9 t_z \\ 1 \end{pmatrix} \quad (2.4)$$

Le système (2.4) s'écrit :

$$\tilde{P} = \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \rightarrow \tilde{P}_c = \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} r_1 & r_2 & r_3 & -r_1 t_x - r_2 t_y - r_3 t_z \\ r_4 & r_5 & r_6 & -r_4 t_x - r_5 t_y - r_6 t_z \\ r_7 & r_8 & r_9 & -r_7 t_x - r_8 t_y - r_9 t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \quad (2.5)$$

d'où

$$\tilde{P} = \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \rightarrow \tilde{P}_c = \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} r_1 & r_2 & r_3 & 0 \\ r_4 & r_5 & r_6 & 0 \\ r_7 & r_8 & r_9 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \quad (2.6)$$

Le système (2.6) peut s'écrire :

$$\tilde{P} = \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \rightarrow \tilde{P}_c = \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R} & 0 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \quad (2.7)$$

avec \mathbf{I} est la matrice identité (3x3) et $\mathbf{0}$ le vecteur nul,

$$\text{d'où,} \quad \tilde{P} = \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \rightarrow \tilde{P}_c = \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \mathbf{M} \tilde{P} \quad (2.8)$$

$$\text{avec} \quad \mathbf{M} = \begin{bmatrix} \mathbf{R} & 0 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Calcul de la matrice K

La figure (2.6) montre la projection du point P_c sur le plan image, le calcul de la matrice \mathbf{K} peut être effectué géométriquement :

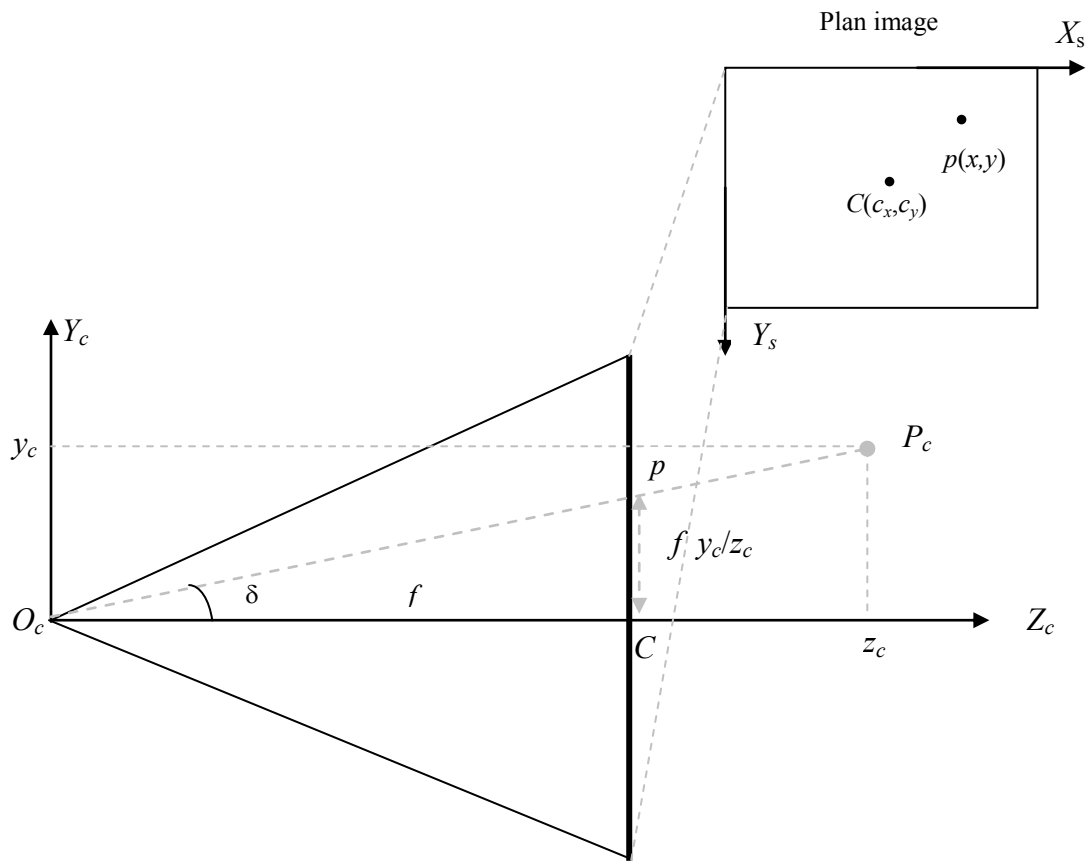


Figure 2.6. : Projection d'un point P sur le plan image.

Le calcul de la tangente de l'angle δ (Figure 2.6) permet d'écrire :

$$\frac{y - c_y}{f} = \frac{y_c}{z_c} \quad (2.9)$$

d'où
$$y = f \frac{y_c}{z_c} + c_y \quad (2.10)$$

Le même raisonnement avec l'axe X_c et l'axe X_s permet de déduire que :

$$x = f \frac{x_c}{z_c} + c_x \quad (2.11)$$

De même que pour le calcul de \mathbf{M} , on écrit les points P_c et p dans l'espace projectif, ainsi le transfert dans l'espace euclidien, de la caméra vers l'image :

$$P_c = (x_c, y_c, z_c) \rightarrow p = (x, y) = \left(f \frac{x_c}{z_c} + c_x, f \frac{y_c}{z_c} + c_y\right) \quad (2.12)$$

devient dans l'espace projectif:

$$\tilde{P}_c = (x_c, y_c, z_c, 1) \rightarrow \tilde{p} = (x, y, 1) = \left(f \frac{x_c}{z_c} + c_x, f \frac{y_c}{z_c} + c_y, 1\right) \quad (2.13)$$

d'où,

$$\tilde{P}_c = (x_c, y_c, z_c, 1) \rightarrow \tilde{p} = (x, y, 1) = (fx_c + z_c c_x, fy_c + z_c c_y, z_c) \quad (2.14)$$

On peut écrire (2.13) sous forme matricielle :

$$\tilde{P}_c = \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} \rightarrow \tilde{p} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} \quad (2.15)$$

$$\text{d'où} \quad \tilde{p} = \mathbf{K} [\mathbf{I} \mid \mathbf{0}] \tilde{P}_c \quad (2.16)$$

$$\text{avec} \quad \mathbf{K} = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{I} \text{ la matrice identité (3x3)} \quad \text{et} \quad \mathbf{0} \text{ vecteur nul.}$$

\mathbf{K} est la matrice de calibrage de la caméra. Elle est formée des paramètres internes de la caméra : la focale f et les coordonnées du point principal C .

L'équation (2.8) et (2.16) permettent d'écrire :

$$\tilde{p} = \mathbf{K} [\mathbf{I} \mid \mathbf{0}] \begin{bmatrix} \mathbf{R} & 0 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tilde{P} \quad (2.17)$$

$$\text{d'où,} \quad \tilde{p} = \mathbf{K} \mathbf{R} [\mathbf{I} \mid -\mathbf{t}] \tilde{P} \quad (2.18)$$

L'équation (2.18) exprime la projection d'un point du monde sur le plan image selon le modèle perspective et en utilisant les paramètres intrinsèques et extrinsèques de la caméra. Ces calculs nous seront utiles dans la génération des images de simulation.

5- Extraction des indices visuels de l'image

Les images brutes sont constituées de points dont la plupart ne contient pas une information pertinente sur la scène, du point de vue de l'analyse. Il est nécessaire alors de procéder à l'extraction d'indices visuels pertinents. Nous avons choisi de travailler avec des points d'intérêt (keypoints) comme indices visuels. Notre choix se justifie par le fait que des points caractéristiques existent sur la plupart des images et qu'ils sont simples à détecter car cela ne nécessite pas d'opération de chaînage (suivi de contours), en plus de disposer d'algorithmes puissants pour l'extraction et la mise en correspondance des points d'intérêt.

L'extraction des points d'intérêt doit se faire sans perte d'information utile, d'où la nécessité de certains critères de choix auxquels doivent répondre ces points tels que : la robustesse au bruit et aux changements de luminosité; la précision de leur position dans l'image ou dans l'objet; l'invariance aux transformations de rotation, d'échelle, de translation,...; la localité qui permet à un point d'être plus robuste aux occlusions; la distinctivité qui permet de mettre correctement en correspondance un point d'intérêt avec un autre point d'intérêt sur une autre image.

Plusieurs types de points d'intérêt sont disponibles dans la littérature [32][41][42][45][46]. Dans ce présent travail, nous utilisons des points SURF (Speeded Up Robust Features) [13] [14], utilisés dans la fonction *ImageKeyPoints* du langage de programmation Mathematica [6]. Cette fonction a pour effet d'extraire les points d'intérêt sur une image donnée, par la méthode SURF et les renvoie sous forme de liste de points 2D. L'algorithme SURF est basé sur l'algorithme SIFT (Scale-Invariant Features Transform) [41][42].

L'algorithme SIFT transforme les données d'une image (valeurs d'intensité des pixels) en données relativement invariantes à l'échelle et aux rotations et rapportées à des caractéristiques locales. La première étape de l'algorithme est la détection des points d'intérêt. La seconde concerne le calcul des descripteurs de ces points d'intérêt.

5-1- Détection des points d'intérêt

L'image originale I est convoluée par une série de k fonctions gaussiennes G de paramètres respectifs $(\sigma_1 = \sigma, \dots, \sigma_k = k\sigma)$ (généralement $k=5$, valeur expérimentale).

$$L(x, y, \sigma_i) = G(x, y, \sigma_i) * I(x, y) \quad i = 1, \dots, k \quad (2.19)$$

En lissant l'image originale I , les détails trop petits (de rayon inférieur à σ_i) sont estompés. Par conséquent, la détection des objets de dimension approximativement égale à σ_i se fait en étudiant l'image obtenue par les différences de gaussiennes (DoG) définie comme suit :

$$D(x, y, \sigma_i) = L(x, y, \sigma_{i+1}) - L(x, y, \sigma_i) \quad i = 1, \dots, k-1 \quad (2.20)$$

Le même calcul des DoG est repris de manière pyramidale avec des échelles différentes. Chaque image L est successivement réduite en doublant le facteur d'échelle à chaque niveau. Chaque niveau regroupant k images gaussiennes représente une octave. On obtient une pyramide de calcul illustrée par la figure (Figure 2.7).

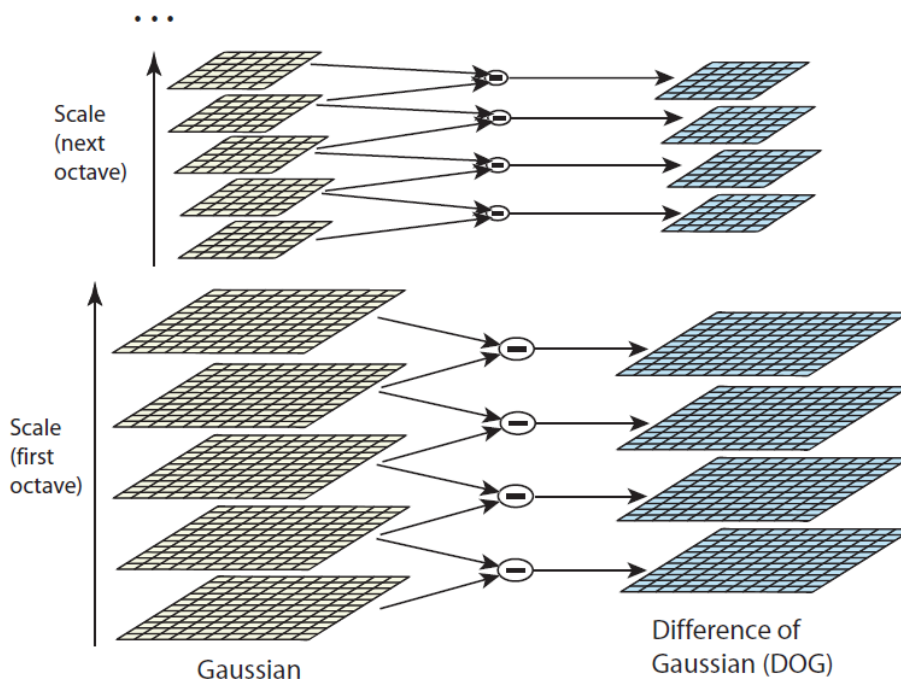


Figure 2.7: Calcul des points SIFT : Pyramide des calculs à 2 octaves [42].

(gauche) Résultats des convolutions avec des gaussiennes à différentes échelles,
(droite) DoG à différentes échelles.

Un point d'intérêt est détecté s'il est un extrema local de la différence de gaussienne par rapport à ses voisins directs (pour une même octave) et aussi par rapport à ses voisins dans les images d'échelle consécutive. La liste des points d'intérêt est filtrée en éliminant d'abord les points mal contrastés et les points se trouvant sur des lignes. Les points sont par la suite localisés avec précision en résolvant un polynôme quadratique basé sur une expansion de Taylor de la DoG dans laquelle le point est détecté.

5-2- Calcul des descripteurs

Pour chaque point d'intérêt, on doit calculer son descripteur, celui-ci stocke des informations sur l'échelle, l'orientation, l'intensité. Chaque zone contenant un point d'intérêt est subdivisée en 4x4 sous régions de 4x4 pixels chacune. Pour chaque sous région, on calcule un histogramme des orientations du gradient (8 intervalles de 45°), ce qui donne une taille du vecteur descripteur de 4x4x8 (Figure 2.8). Une pondération est effectuée en fonction de la distance au centre et de la magnitude du gradient afin de déterminer l'orientation principale qui dépend du contenu local de l'image autour du point d'intérêt. Le calcul de l'orientation permet d'assurer l'invariance de la méthode à la rotation.

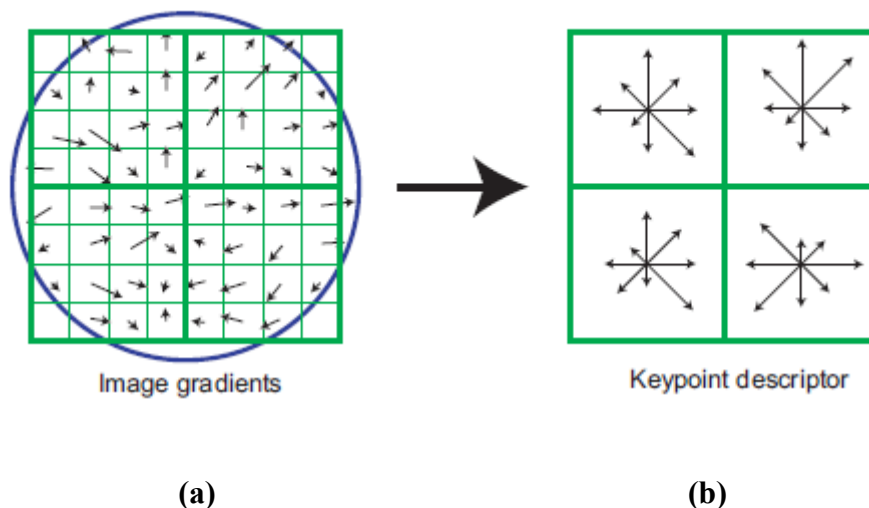
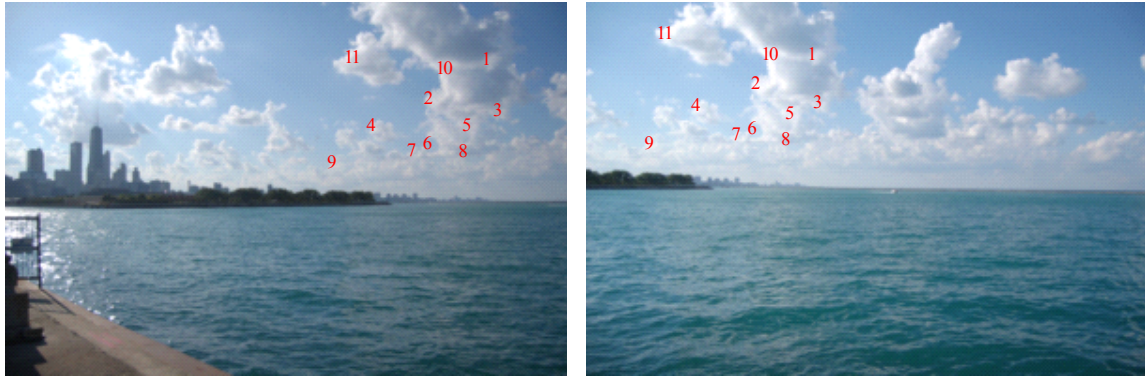


Figure 2.8 : Calcul du descripteur d'un point SIFT [42]

(a) Gradients de la zone contenant un point d'intérêt.

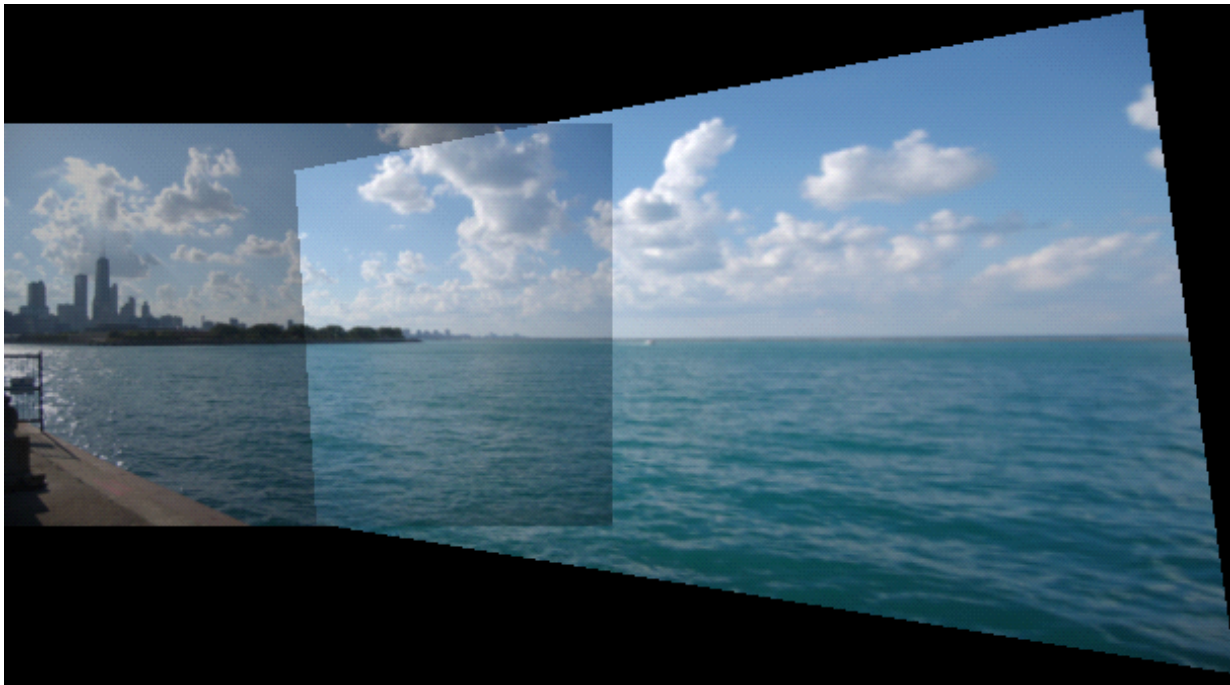
(b) Descripteur du point d'intérêt.

La figure 2.9 montre l'invariance des points SIFT par rapport à la translation, à l'échelle et à la rotation :



(a) Image₁

(b) Image₂



(c)

Figure 2.9 : Invariance des points SIFT à la translation, aux changements d'échelle et à la rotation pour la construction d'un panorama.

(a) Image₁ avec les points SIFT mis en correspondance avec Image₂

(b) points SIFT correspondants sur Image₂ (c) construction du panorama par transformation de Image₂ et fusion avec Image₁.

Contrairement à l'algorithme SIFT et afin d'assurer une accélération dans les calculs, l'algorithme SURF (Speeded Up Robust Features) ou (indices visuels robustes accélérés) utilise des images intégrales pour calculer les convolutions. Une image intégrale est une représentation sous forme d'image, de même taille que l'image originale, où chacun de ses pixels représente la somme des intensités des pixels situés au dessus et à gauche de celui-ci. En plus, SURF utilise une approximation d'ondelettes de Haar dans le voisinage du point d'intérêt, à base de déterminant Hessien [13][14]. La dimension du descripteur est réduite de 128 à seulement 64 ce qui permet d'accélérer davantage le temps de calcul.

6- Mise en correspondance de deux images

La mise en correspondance des points d'intérêt appartenant à deux images successives détermine pour chaque point d'intérêt p_1 sur Image₁ son homologue le point p_2 s'il existe sur Image₂, tels que p_1 et p_2 soient la projection d'un même point P de la scène 3D, sur Image₁ et Image₂ respectivement (Figure 2.10). Cette mise en correspondance se fait en se basant sur le calcul d'une distance ϕ telle que distance euclidienne, distance de Mahalannobis, distance algébrique,... entre les deux descripteurs d_1 et d_2 représentant respectivement les deux points d'intérêt p_1 et p_2 .

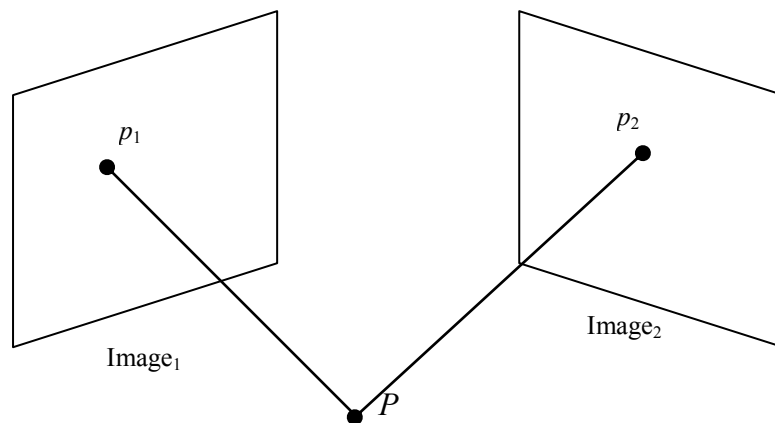


Figure 2.10 : Mise en correspondance de deux points images.

La mise en correspondance de deux points images peut présenter des ambiguïtés. Afin de pallier à ce problème et éliminer les fausses cibles, certaines contraintes ont été formulées par Marr et Poggio [43][53][56] :

- **Contrainte d'unicité**

Un point de Image₁ a au plus un homologue dans Image₂. Cette contrainte exclut la possibilité d'alignement des points 3D avec le centre optique de la caméra.

- **Contrainte de continuité**

Le monde physique étant généralement constitué de surfaces continues, cette contrainte permet à partir d'une mise en correspondance initiale de prédire d'autres mises en correspondance qui viennent confirmer la première. Skordas et Horaud [56] ont énoncé les contraintes suivantes :

- **Contrainte de continuité de disparité** : La disparité métrique est définie par :

$$\lambda(P) = p_1 C_1 - p_2 C_2 \quad (2.21)$$

C'est la différence des positions des points p_1 et p_2 par rapport aux points principaux respectivement sur Image₁ et Image₂. La contrainte de continuité de disparité s'appuie sur le fait que les surfaces physiques sont localement continues et que la variation de leur profondeur par rapport à l'observateur est généralement faible.

- **Contrainte de continuité des formes** : Une suite de points de contours connectés dans Image₁ devrait apparaître comme une séquence connectée de points de contours dans Image₂.

- **Contrainte de ressemblance**

Deux primitives sont mises en correspondance si leurs caractéristiques vérifient les conditions de ressemblance entre eux. Ces conditions peuvent être d'ordre géométrique ou photométrique.

- **Contrainte d'ordre**

Cette contrainte suppose que l'ordre dans lequel sont disposées des primitives appartenant à Image_1 est celui dans lequel leurs homologues sont disposés.

L'opération de mise en correspondance des points d'intérêt sur les deux images est conditionnée par les contraintes citées ci-dessus. Elle a pour résultat de fournir une liste de paires de points homologues appartenant respectivement aux deux images.

6-1- Stratégies de mise en correspondance

Plusieurs stratégies peuvent être utilisées dans la mise en correspondance de deux points d'intérêt :

Deux points d'intérêt sont classés correspondants si la distance ϕ (que l'on choisit) entre leurs descripteurs est inférieure à un seuil ξ .

$$\phi(d_1, d_2) < \xi \quad (2.22)$$

Le choix du seuil influe sur la performance de la mise en correspondance. En plus, cette stratégie de mise en correspondance peut être ambiguë si le point p_1 a un plus proche voisin p_2 et un second plus proche voisin p_2' . Dans ce cas, une autre stratégie peut être utilisée. Elle se base sur le calcul du ratio des distances $\phi(d_1, d_2)$ et $\phi(d_1, d_2')$ associées respectivement aux points p_2 et p_2' . Si ce ratio est suffisamment grand, le plus proche voisin est accepté comme correspondant à p_1 .

De telles stratégies donnent un ensemble préliminaire de correspondances qui peut contenir des correspondances aberrantes. Elles peuvent être améliorées par le calcul homographique si ces points appartiennent à un même plan. Le calcul homographique peut à son tour être affiné en utilisant la méthode RANSAC [25].

7- Calcul homographique

L'opération de mise en correspondance des points d'intérêt sur les deux images fournit une liste de paires de points homologues. Étant donné que la caméra est en rotation pure, on

considère qu'il existe une transformation projective représentée par une matrice \mathbf{H} , reliant les points mis en correspondance sur les deux images successives.

Rappelons que l'équation (2.18) $\tilde{p} = \mathbf{K} \mathbf{R} [\mathbf{I} \quad -\mathbf{t}] \tilde{P}$ permet d'exprimer la projection d'un point du monde \tilde{P} sur un plan image d'une caméra de matrice de calibrage \mathbf{K} et dont le repère est positionné, par rapport au centre du monde, avec une rotation \mathbf{R} et une translation \mathbf{t} .

Lorsque la caméra est en position initiale, la projection du point \tilde{P} sur Image₁ s'écrit selon l'équation (2.18) :

$$\tilde{p}_1 = \mathbf{K} \mathbf{R}_1 [\mathbf{I} \quad -\mathbf{t}_1] \tilde{P}$$

\mathbf{R}_1 est une matrice de rotation, elle est donc inversible.

\mathbf{K} est une matrice inversible, car elle est triangulaire et tous les éléments diagonaux sont non nuls,

$$\text{d'où} \quad \mathbf{R}_1^{-1} \mathbf{K}^{-1} \tilde{p}_1 = [\mathbf{I} \quad -\mathbf{t}_1] \tilde{P} \quad (2.23)$$

Lorsque cette même caméra fait une rotation pure par rapport à sa position initiale, c-à-d sans translation par rapport à sa position initiale, la projection du point \tilde{P} sur Image₂ s'écrit selon l'équation (2.18) :

$$\tilde{p}_2 = \mathbf{K} \mathbf{R}_2 [\mathbf{I} \quad -\mathbf{t}_1] \tilde{P}$$

d'après (2.23), on peut écrire :

$$\tilde{p}_2 = \mathbf{K} \mathbf{R}_2 \mathbf{R}_1^{-1} \mathbf{K}^{-1} \tilde{p}_1 \quad (2.24)$$

$$\text{d'où} \quad \tilde{p}_2 = \mathbf{H} \tilde{p}_1 \quad \text{avec} \quad \mathbf{H} = \mathbf{K} \mathbf{R}_2 \mathbf{R}_1^{-1} \mathbf{K}^{-1} \quad (2.25)$$

Cette transformation projective (2.25) appelée homographie est une transformation linéaire non singulière pour des coordonnées homogènes. Elle est donc, définie dans un espace projectif. Celui-ci est l'espace euclidien augmenté des points à l'infini. Les points à l'infini appelés aussi points idéaux permettent d'envisager entre autres, la projection des lignes parallèles qui se coupent à l'infini dans une scène 3D.

Afin d'exploiter la linéarité de cette transformation projective, il est nécessaire de transformer les points images initialement définis dans l'espace euclidien en des points

projectifs. Ainsi les différents points de coordonnées (x, y) de l'espace euclidien seront transformés dans l'espace projectif par l'ajout d'une troisième coordonnée homogène :

La transformation d'un point de l'espace euclidien vers l'espace projectif se fait par :

$$p = \begin{pmatrix} x \\ y \end{pmatrix} \longrightarrow \tilde{p} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \text{ou bien} \quad \tilde{p} = \begin{pmatrix} kx \\ ky \\ k \end{pmatrix} \quad \text{avec } k \neq 0 \quad (2.26)$$

La transformation d'un point de l'espace projectif vers l'espace euclidien se fait par :

$$\tilde{p} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \longrightarrow p = \begin{pmatrix} x/z \\ y/z \end{pmatrix} \quad \text{avec } z \neq 0 \quad (2.27)$$

Les ensembles des points p_1 et p_2 respectivement de Image_1 et de Image_2 forment dans l'espace projectif deux plans π_1 et π_2 contenant les points projectifs \tilde{p}_1 et \tilde{p}_2 correspondants à p_1 et p_2 .

Dans l'espace projectif, l'homographie reliant les deux plans π_1 et π_2 est représentée par une matrice $\mathbf{H}_{3 \times 3}$ reliant deux points $\tilde{p}_1(x_1, y_1, 1)$ et $\tilde{p}_2(x_2, y_2, 1)$ appariés appartenant respectivement à π_1 et π_2 :

$$\tilde{p}_2 = \mathbf{H} \tilde{p}_1 \quad (2.28)$$

d'où

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \quad (2.29)$$

Il est à noter que \mathbf{H} peut varier en la multipliant par une constante arbitraire non nulle, sans que cela n'altère la transformation projective. \mathbf{H} peut donc être considérée comme une matrice avec 8 degrés de liberté même si elle contient 9 éléments. Cela signifie que pour retrouver la matrice \mathbf{H} , il suffit de déterminer les 8 inconnues au lieu de 9, si on prend $h_9 = 1$. Hartley et al [33] proposent une méthode d'estimation de l'homographie \mathbf{H} .

7-1- Estimation de l'homographie [24][33]

7-1-1- L'algorithme DLT

L'algorithme DLT (Direct Linear Transform) permet d'estimer l'homographie \mathbf{H} .

Le développement de (2.29) donne :

$$\begin{aligned}h_1x_1 + h_2y_1 + h_3 &= x_2 \\h_4x_1 + h_5y_1 + h_6 &= y_2 \\h_7x_1 + h_8y_1 + h_9 &= 1\end{aligned}\tag{2.30}$$

En divisant la ligne 1 par la ligne 3 et la ligne 2 par la ligne 3, dans (2.30), on obtient :

$$\begin{aligned}h_1x_1 + h_2y_1 + h_3 - x_2(h_7x_1 + h_8y_1 + h_9) &= 0 \\h_4x_1 + h_5y_1 + h_6 - y_2(h_7x_1 + h_8y_1 + h_9) &= 0\end{aligned}\tag{2.31}$$

Les équations (2.31) peuvent être écrites sous la forme :

$$\mathbf{A}_j \mathbf{h} = 0\tag{2.32}$$

avec j : $j^{\text{ème}}$ paire de points mis en correspondance.

$$\text{où } \mathbf{A}_j = \begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_2 & -y_1x_2 & -x_2 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_2 & -y_1y_2 & -y_2 \end{pmatrix}\tag{2.33}$$

$$\text{et } \mathbf{h} = (h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ h_7 \ h_8 \ h_9)^T\tag{2.34}$$

Chaque paire de points mis en correspondance fournit 2 équations dans la matrice \mathbf{A}_j . Donc, 4 paires de points sont suffisants pour résoudre le système à 8 degrés de liberté de \mathbf{H} . En regroupant les 4 paires de lignes \mathbf{A}_j ($j=1,\dots,4$), on obtient le système :

$$\mathbf{A} \mathbf{h} = 0\tag{2.35}$$

où \mathbf{A} est une matrice 8×9 .

Un nombre de points supérieur à 4 permet de donner plus de robustesse à la solution. Dans ce cas, la solution du système (2.35) est surdéterminée, sa résolution revient à trouver \mathbf{h} qui minimise une fonction de coût à définir. On peut calculer la solution des moindres carrés $\|\mathbf{A}\mathbf{h}\|$, qui consiste à retrouver le vecteur singulier ayant la plus petite valeur singulière. Cela peut se faire par la méthode SVD (Singular Value Decomposition) qui minimise la distance algébrique d_{alg} (2.36). Celle-ci est la norme du produit vectoriel entre un point et sa reprojection par \mathbf{H} :

$$d_{\text{alg}}(\tilde{p}_1, \tilde{p}_2, \mathbf{H}) = \|\tilde{p}_2 \times \mathbf{H}\tilde{p}_1\| \quad (2.36)$$

Afin d'obtenir un vecteur \mathbf{h} non nul, on minimise la distance algébrique avec la contrainte $\|\mathbf{h}\|=1$.

La factorisation SVD permet d'écrire la matrice rectangulaire \mathbf{A} sous forme de produit de trois matrices : une matrice orthogonale \mathbf{Q} dont les colonnes sont formées des vecteurs propres de $\mathbf{A}\mathbf{A}^T$, une matrice \mathbf{L} contenant des racines carrées des valeurs propres de $\mathbf{A}\mathbf{A}^T$ disposées par ordre décroissant de leurs valeurs et une matrice transposée d'une matrice orthogonale \mathbf{U} . Les colonnes de \mathbf{U} sont formées des vecteurs propres de $\mathbf{A}^T\mathbf{A}$. La matrice \mathbf{A} est décomposée en :

$$\mathbf{A}_{mn} = \mathbf{Q}_{mm} \mathbf{L}_{mn} \mathbf{U}_{nn}^T \quad (2.37)$$

Le vecteur \mathbf{h} qu'on cherche à déterminer, correspond au dernier vecteur propre de la matrice \mathbf{U} qui lui-même correspond à la plus petite valeur propre de $\mathbf{A}\mathbf{A}^T$.

7-1-2- Normalisation des pixels

La solution du système à huit équations (2.35) est lié à l'origine et à l'échelle des coordonnées des points [24][33], car les pixels dans l'espace projectif ont des coordonnées x et y beaucoup plus grandes que 1 qui est la troisième coordonnée homogène. Cela engendre de l'instabilité numérique. Afin de pallier à cet inconvénient, Hartley et al [33] présentent l'algorithme DLT avec normalisation des pixels (Algorithme 2.1).

Entrée : liste des paires de points mis en correspondance.

Sortie : **H** homographie entre les points mis en correspondance.

Début

1- Déterminer pour chaque image i ($i=1, 2$), une transformation similarité (translation et changement d'échelle) $T_{i(3 \times 3)}$ qui permet de centrer les points d'intérêt à l'origine (leur centroïde devient l'origine) à une distance moyenne de l'origine égale à $\sqrt{2}$.

2- Calculer par SVD une transformation **H'** à partir des paires de points normalisés obtenus en 1.

3- L'homographie **H** est calculée par : $H = T_2^{-1} H' T_1$

Fin.

Algorithme 2.1 : Calcul de H par DLT avec normalisation des pixels [33].

L'estimation de l'homographie par la méthode SVD minimise l'erreur algébrique et la normalisation des pixels permet d'améliorer la solution obtenue par SVD (Algorithme 2.1). Cependant cette solution linéaire n'est pas optimale, on peut alors approcher davantage la solution optimale, en utilisant la méthode itérative Ransac dans l'estimation de l'homographie **H** et par conséquent, dans l'estimation des vraies correspondances.

7-1-3- Optimisation de l'estimation de l'homographie par Ransac [24][33]

Ransac est une méthode itérative robuste qu'on peut utiliser dans l'estimation de l'homographie [24][33]. Parmi l'ensemble S des paires de points mis en correspondance, on sélectionne aléatoirement un ensemble \hat{S} de 4 correspondances (4 paires de points) et on calcule l'homographie **H** les reliant selon l'algorithme 2.1 qui se base sur la distance algébrique. Les autres correspondances seront soumises à cette homographie et on calcule pour chaque correspondance la distance euclidienne entre un point \tilde{p}_2 et sa reprojection par **H** :

$$d_{eucl}(\tilde{p}_1, \tilde{p}_2, \mathbf{H}) = \|\tilde{p}_2 - \mathbf{H} \tilde{p}_1\| \quad (2.38)$$

Une correspondance est acceptée, donc classée comme vraie, si le carré de la distance euclidienne entre les deux points appariés est inférieure à un seuil τ :

$$d_{eucl}(\tilde{p}_1, \tilde{p}_2, \mathbf{H}) < \tau \quad (2.39)$$

Chaque correspondance acceptée est stockée dans l'ensemble \hat{S}_n . Le calcul est réitéré N fois (§ 7-1-4) et à chaque itération on retient l'ensemble \hat{S}_n ayant donné le plus grand nombre de vraies correspondances. Il devient l'ensemble référence S^* , dans le sens où il contient le plus grand nombre de correspondances acceptées, pour la prochaine itération. À la fin, on recalcule \mathbf{H} pour une dernière fois selon l'algorithme 2.1 sur l'ensemble des correspondances retenues S^* .

7-1-4- Calcul du nombre d'itérations N [24]

Le nombre d'itérations N est déterminé en se basant sur un calcul de probabilité :
 Soit ε la proportion des fausses correspondances (les paires de points faussement appariés).
 La probabilité qu'une correspondance soit vraie est $\omega = 1 - \varepsilon$.
 La probabilité de sélectionner s vraies correspondances est ω^s .
 La probabilité d'avoir au moins une fausse correspondance parmi s correspondances est $1 - \omega^s$.
 La probabilité d'avoir au moins une fausse correspondance dans N choix de sous ensembles est $(1 - \omega^s)^N$.
 La probabilité d'avoir au moins un choix de s correspondances sans fausses correspondances après N essais est: $\nu = 1 - (1 - \omega^s)^N$.
 Le nombre d'essais nécessaires pour obtenir une probabilité ν d'avoir fait au moins un choix de s éléments sans fausse correspondance est :

$$N = \frac{\ln(1 - \nu)}{\ln(1 - (1 - \varepsilon)^s)} \quad (2.40)$$

La valeur de la probabilité est choisie $\nu=0.99$, afin de maximiser les chances des vraies correspondances. La valeur de s est 4, c'est le minimum de correspondances nécessaires pour le calcul de l'homographie. La valeur du seuil τ est choisie par expérimentation.

L'estimation de l'homographie par Ransac est réalisée par l'algorithme 2.2 :

Entrée: $S = \{ (\tilde{p}_{1j}, \tilde{p}_{2j}) \mid j = 1, \dots, |S| \}$, ensemble des points appariés par l'algorithme 2.1

Sortie : S^* ensemble des paires de points formant des correspondances acceptées.

H : l'homographie estimée.

Paramètres intermédiaires : N (nombre maximal d'itérations), \hat{S} (ensemble de s correspondances choisies aléatoires), \hat{S}_n (ensemble des correspondances acceptées à l'itération n), ε (proportion des fausses correspondances)

Début :

$N \leftarrow \infty$, $n=1$, $S^* \leftarrow \emptyset$ // Initialisation

Tant que $n < N$ **faire**

Choisir aléatoirement $\hat{S} \subset S$ tel que $|\hat{S}| = s$

Estimer **H** sur \hat{S} (selon l'algorithme 2.1)

$\hat{S}_n \leftarrow \{ (\tilde{p}_{1j}, \tilde{p}_{2j}) \mid (\tilde{p}_{1j}, \tilde{p}_{2j}) \in S \text{ et } d_{eucl}(\tilde{p}_{1j}, \tilde{p}_{2j}, \mathbf{H}) < \tau \}$

Si $|\hat{S}_n| > |S^*|$ **alors**

$S^* \leftarrow \hat{S}_n$

$\varepsilon \leftarrow 1 - |\hat{S}_n| / |S|$

$N \leftarrow \frac{\ln(1 - \nu)}{\ln(1 - (1 - \varepsilon)^s)}$

Fin Si

$n \leftarrow n+1$

fin Tant que

Estimer **H** sur S^*

Fin

Algorithme 2.2 : Estimation de l'homographie par RANSAC adaptatif [24][25][33].

Le résultat de l'algorithme 2.2 retient les points les plus stables au sens de l'homographie et la convergence semble être assurée empiriquement. L'étape suivante est de déterminer les points en mouvement.

8- Détection de mouvement

8-1- Calcul des points en mouvement

L'estimation de l'homographie et la détermination des vraies correspondances par la méthode Ransac permet de retenir les points d'intérêt ayant été projetés sur les deux images par des points de la scène. Ces points présentent le meilleur score en termes de correspondance, entre les deux images, donc en termes de stabilité dans la scène, puisque Ransac converge vers la meilleure solution [25]. Nous exploitons cette propriété pour extraire les points en mouvement sur Image₂ :

Un point \tilde{p}_2 sur Image₂ est considéré en mouvement s'il ne satisfait pas l'homographie \mathbf{H} , c'est-à-dire s'il n'est 2^{ème} composante d'aucune paire de points de l'ensemble S^* obtenu par Algorithme 2.2. Le point \tilde{p}_2 est classé en mouvement si :

$$\tilde{p}_2 \neq \tilde{p}_{2j}, \quad \forall (\tilde{p}_{1j}, \tilde{p}_{2j}) \in S^*, j=1, \dots, |S^*| \quad (2.41)$$

L'algorithme de détection des points en mouvement est le suivant :

Entrée : Image₁ et Image₂

Sortie : V : ensemble des points en mouvement dans Image₂
 B : ensemble des points statiques dans Image₂

Début

$V \leftarrow \emptyset; B \leftarrow \emptyset;$

Pour chaque paire d'images successives Image₁ et Image₂ **faire**

 1- Extraire les points d'intérêt dans les deux images.

 2- Déterminer toutes les correspondances $(\tilde{p}_1, \tilde{p}_2)$ entre les deux images.

 3- Calculer \mathbf{H} et S^* selon l'algorithme 2.2 (retenir les correspondances statiques)

$$S^* = \{ (\tilde{p}_{1j}, \tilde{p}_{2j}) \mid j = 1, \dots, |S^*| \}$$

 4- **Pour** tout point \tilde{p}_2 déterminé en (2) **faire**

Si $\tilde{p}_2 \neq \tilde{p}_{2j}, \quad \forall (\tilde{p}_{1j}, \tilde{p}_{2j}) \in S^*, j=1..|S^*|$

alors $V \leftarrow V \cup \{ \tilde{p}_2 \}$ **sinon** $B \leftarrow B \cup \{ \tilde{p}_2 \}$

fin pour;

fin pour;

Fin.

Algorithme 2.3: Détection des points en mouvement.

L'ensemble V des points en mouvement obtenu par l'algorithme 2.3, peut contenir des faux positifs. En effet un point statique dans la scène peut être projeté sur une seule image des deux images consécutives, donc un point statique n'aurait pas eu son correspondant par l'algorithme 2.2. Ceci peut se produire pour différentes raisons :

- 1- Un point se trouve à la limite de Image₁ peut disparaître sur Image₂ parce que la caméra effectue des rotations.
- 2- L'algorithme 2.2 est une estimation de l'homographie et une classification des correspondances. Il n'est pas exclu que des erreurs de classification influent le résultat.

Pour ces raisons, nous proposons d'éliminer les points statiques ayant été classés en mouvement (c-à-d les faux positifs), en estimant le mouvement par une méthode combinant une méthode de classification basée sur un calcul de distance et la méthode Ransac.

8-2- Élimination des faux positifs

1^{ère} étape :

Pour tout point classé en mouvement par l'algorithme 2.3, si la distance euclidienne minimale qui le sépare des points classés statiques (c-à-d les points rejetés par l'algorithme 2.3) est inférieure à un seuil τ' , alors il sera considéré comme un point statique et sera éliminé de l'ensemble des points en mouvement. L'ensemble V obtenu par l'algorithme 2.3 sera ainsi réduit, il sera noté V' .

2^{ème} étape :

Parmi les points de l'ensemble V' obtenu à l'étape précédente, nous sélectionnons aléatoirement un ensemble \hat{V} de s points et nous calculons l'isobarycentre $g = (x_g, y_g)$ de \hat{V} . Pour chacun des autres points de V' , nous calculons la distance euclidienne :

$$d_{eucl}(p_2, g) = \|p_2 - g\| \quad (2.42)$$

Un point p_2 est confirmé en mouvement si la distance euclidienne entre ce point et g est inférieure à un seuil τ'' :

$$d_{eucl}(p_2, g) < \tau'' \quad (2.43)$$

Cette inéquation modélise le fait que nous recherchons des points d'intérêt en mouvement appartenant à une forme mobile compacte.

Chaque point confirmé en mouvement est stocké dans l'ensemble \hat{V}_n . Le calcul est réitéré N' fois et à chaque itération on retient l'ensemble \hat{V}_n ayant donné le maximum de points confirmés en mouvement, il devient l'ensemble référence V^* , pour la prochaine itération.

Pour le calcul du nombre d'itération, le même raisonnement est appliqué que pour l'équation (2.40) :

Soit ε' la proportion des points ayant été faussement classés en mouvement.

La probabilité qu'une classification en mouvement soit vraie est $\omega' = 1 - \varepsilon'$.

La probabilité de sélectionner s vraies points en mouvement est $(\omega')^s$.

La probabilité d'avoir au moins un faux positif parmi s point de V' est $1 - (\omega')^s$.

La probabilité d'avoir au moins un faux positif dans N' choix de sous ensembles est :

$$(1 - (\omega')^s)^{N'}.$$

La probabilité d'avoir au moins un choix de s points de V' sans faux positifs après N' essais est: $\nu' = 1 - (1 - (\omega')^s)^{N'}$.

Le nombre d'essais nécessaires pour obtenir une probabilité ν' d'avoir fait au moins un choix de s éléments sans faux positifs est :

$$N' = \frac{\ln(1 - \nu')}{\ln(1 - (1 - \varepsilon')^s)} \quad (2.44)$$

L'algorithme qui permet de retenir les points en mouvement est :

Entrée: $V = \{p_{2j} \mid j=1,\dots,|V|\}$, ensemble des points classés en mouvement par l'algorithme 2.3

B = ensemble des points classés statiques par l'algorithme 2.3

Sortie : V^* ensemble des points en mouvement acceptés.

Début :

1^{ère} étape :

$V' \leftarrow V$

Pour tout point p_{2j} de V

faire

Si $\exists q \in B$ tel que $d_{eucl}(p_{2j}, q) < \tau'$

alors $V' \leftarrow V' - \{p_{2j}\}$

fin si

fait

2^{ème} étape : (Ransac)

$N' \leftarrow \infty$, $n=1$, $V^* \leftarrow \emptyset$ // Initialisation

Tant que $n < N'$ **faire**

Choisir aléatoirement $\hat{V} \subset V'$ tel que $|\hat{V}| = s$

$g \leftarrow$ centroïde de \hat{V}

$\hat{V}_n \leftarrow \{p_{2j} \mid p_{2j} \in V' \text{ et } d_{eucl}(p_{2j}, g) < \tau''\}$

Si $|\hat{V}_n| > |V^*|$ **alors**

$V^* \leftarrow \hat{V}_n$

$\varepsilon' \leftarrow 1 - |\hat{V}_n| / |V'|$

$N' \leftarrow \frac{\ln(1 - \nu')}{\ln(1 - (1 - \varepsilon')^s)}$

fin si

$n \leftarrow n+1$

fin tant que

fin

Algorithme 2.4 : Amélioration de la détection des points en mouvement.

9- Détection de chute

Ce travail est dédié aux personnes âgées dont la mobilité est limitée. Nous considérons alors, que tout mouvement brusque entre deux images est une chute, sans tenir compte de sa direction, étant donné qu'il est impossible à une personne âgée dont la mobilité est limitée, de courir.

La détection de chutes se fait par évaluation de la vitesse des points détectés en mouvement par l'algorithme 2.4. Ceux-là forment un nuage de points sur l'image que nous modélisons par un modèle médian. Ainsi, parmi les points de l'ensemble V^* , nous gardons les points dont les coordonnées sont dans l'intervalle du 5^{ème} percentile et du 95^{ème} percentile de l'ensemble V^* . Cela permet de construire une boîte englobante formée des points condensés autour du point médian, ce qui donne plus de robustesse à l'algorithme.

L'évaluation de la vitesse du mouvement se fait par le suivi de la médiane du nuage de points entre les deux images successives. Soient n_1 et n_2 les points médians du nuage des points en mouvements, respectivement sur les deux images successives Image₁ et Image₂. Nous calculons la distance euclidienne entre n_1 et n_2 .

Le mouvement sera classé en mouvement de chute si la distance euclidienne entre n_1 et n_2 est supérieur à un seuil ρ .

$$d_{euc}(n_1, n_2) > \rho \quad (2.45)$$

Calcul du seuil ρ

Afin de prendre en compte des transformations perspectives dans l'image, nous calculons un seuil adaptatif qui est mis à jour en fonction de la position de la personne en mouvement dans la scène 3D. Une personne éloignée de la caméra apparaît plus petite sur l'image que si elle était proche de celle-là. Cet effet de perspective génère sur l'image, un nuage de points condensé, en position éloignée de la caméra et un nuage de points dispersé en

position rapprochée de la caméra (Figure 2.11). Donc un déplacement sur l'image est d'autant plus grand que si la personne est proche de la caméra.

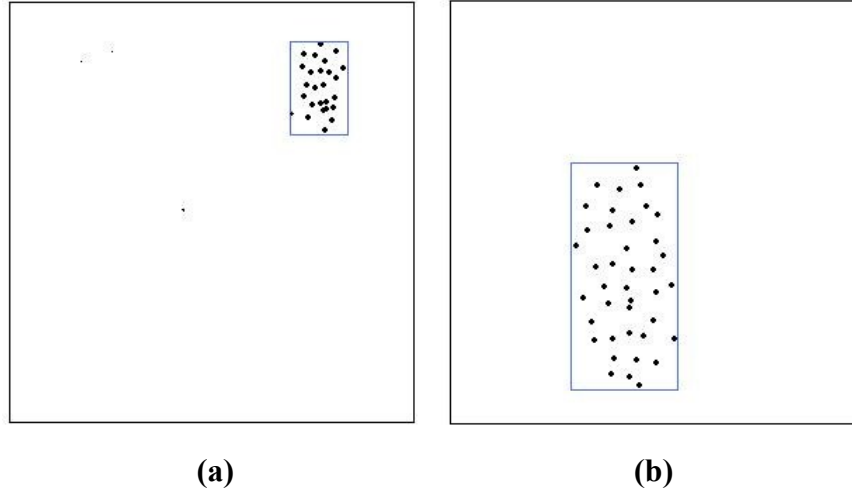


Figure 2.11 : Effet de la transformation perspective sur la taille du nuage de points. (a) Personne éloignée de la caméra. (b) Personne proche de la caméra

Compte tenu de la constance de la vitesse d'un déplacement normal d'une personne âgée, on considère que son déplacement est proportionnel à sa taille. Soit l la hauteur de la boîte englobante calculée sur $Image_1$. Nous considérons que tout déplacement de la médiane du nuage sur $Image_2$ supérieur à $\frac{l}{a}$ est une chute, où a est une constante définie par expérimentation.

Ainsi, un mouvement est classé en chute si :

$$d_{eucl}(n_1, n_2) > \frac{l}{a} \quad (2.46)$$

L'algorithme 2.5 résume la détection de mouvement :

Entrée : V^* et V_{prec}^* ensemble des points en mouvements retenus par l'algorithme 2.4 en 2 itérations successives.

Sortie : Classe du mouvement de V^* .

Début

1- Calcul des boîtes englobantes des 95% des points de V^* et V_{prec}^* .

2- Calcul des médianes n_1 et n_2

3- Si $d_{euc}(n_1, n_2) > \frac{l}{a}$

alors chute

sinon mouvement normal

Fin

Algorithme 2.5 : Détection de chute.

10- Conclusion

Dans ce chapitre, nous avons développé ce qui suit :

Un algorithme de détection de mouvement sur des images prises par une caméra en rotation pure. Cet algorithme se base sur la transformation projective qui relie les deux images successives et opère sans la connaissance du modèle de caméra.

Un algorithme de détection de chute qui se base sur l'évaluation de la vitesse du mouvement relativement à la taille du nuage des points en mouvements entre deux images successives.

Dans le chapitre suivant, nous présentons les détails de l'implémentation, les résultats de l'expérimentation ainsi que leur interprétation.

Chapitre 3

EXPÉRIMENTATION

1- Introduction

L'expérimentation présente un grand intérêt pour le présent travail. En effet, l'algorithme de mise en correspondance entre les points d'intérêt des différentes images et la détection de mouvement par calcul d'homographie sont basés sur une étude théorique, dont il est nécessaire de situer la validité et la robustesse par expérimentation.

Dans une première partie, la méthode est appliquée à des données contrôlées, c-à-d des images que nous synthétisons et où nous contrôlons le mouvement ainsi que les sources de lumière. Ceci nous permet d'évaluer la validité de notre méthode.

Dans une deuxième étape, nous évaluons la robustesse de la méthode en utilisant des images réelles.

2- Environnement de l'expérimentation

2-1- Configuration de la machine

L'expérimentation a été réalisée en environnement Linux, sur une machine du DIRO BLG22, constituée d'un multiprocesseur à 8 unités de 1596 MHz de fréquence et d'une RAM de 6 Go.

2-2- Langage de programmation

Nous avons utilisé le langage de programmation Mathematica version 8 dans un environnement Linux [6]. C'est un logiciel de calcul formel comptant plusieurs bibliothèques notamment des bibliothèques de fonctions mathématiques élaborées dédiées au traitement de l'image.

Afin de générer les images de synthèse, nous avons utilisé POV-Ray (Persistence of Vision Raytracer) [3]. C'est un logiciel à libre usage conçu pour créer des images de synthèse représentant des scènes à deux et trois dimensions, selon des modèles mathématiques. Il peut

être utilisé sous différents systèmes d'exploitation, notamment sous Linux. Les images créées par POV-Ray peuvent être générées sous forme d'une séquence vidéo contenant de l'animation.

3- Données de l'expérimentation

Nous avons utilisées des séquences d'images de 2 types :

- des séquences d'images de synthèse animées,
- des séquences d'images photos prises pour différentes positions de l'entité mobile et avec rotation de l'appareil photo correspondant à des séquences vidéos prises par une caméra en rotation.

3-1- Images de synthèse

La première expérimentation est réalisée sur des images de synthèse générées par le logiciel POV-Ray. Ces images, de type bmp et de dimensions 512 x 384, simulent une scène intérieure élémentaire représentant une pièce d'une maison de dimensions $(L \times l \times H) = (4m \times 4m \times 3m)$.

Les images représentent une scène comprenant un cylindre mobile dans une pièce, simulant grossièrement le déplacement d'une personne âgée.

3-2- Images réelles

Les prises de vue sont réalisées avec un appareil photo de type FinePix F460 de résolution 5.10 Mégapixels. L'appareil est monté sur un trépied lui permettant de faire des rotations. Les images obtenues sont de type JPG et de dimensions 2592 x 1944. Elles représentent une scène intérieure réelle avec un robot à différentes positions, simulant un mouvement de chute et un mouvement de déplacement normal. Nous constituons ainsi une séquence d'images qui se rapproche le plus possible d'une séquence vidéo.

Toutes les images utilisées dans l'expérimentation sont initialement en RGB. Elles sont converties en images à niveaux de gris afin d'alléger les calculs en réduisant le nombre de canaux à 1. Cela n'influe pas sur les résultats de l'opération d'extraction des points d'intérêt, puisque celle-ci est basée sur le calcul du gradient d'intensité.

4- Description du processus de réalisation

Le processus de réalisation est illustré par le schéma suivant :

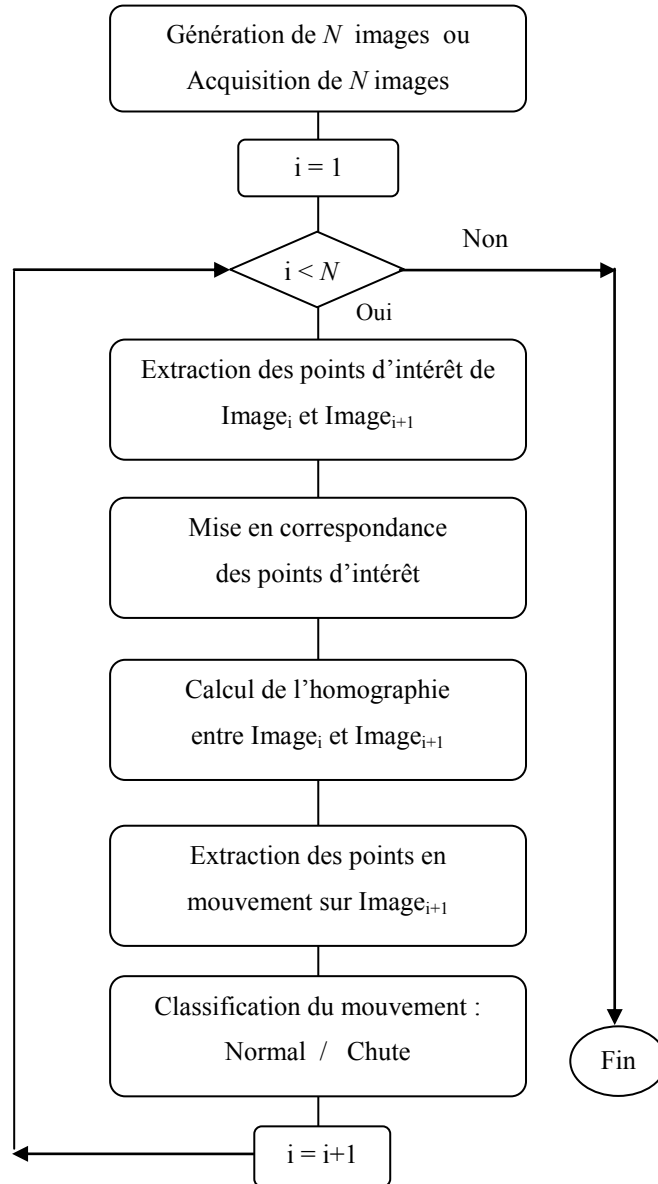


Figure 3.1 : Synoptique des différents traitements.

Dans ce qui suit, nous présentons chacune de ces phases de traitements :

4-1- Génération des Images de synthèse / Acquisition des images réelles

Cette étape consiste à fournir selon le cas, des images de synthèse animées générées par POV-Ray ou bien des images réelles prises à l'aide d'un appareil numérique.

4-2- Extraction des points d'intérêt

L'extraction des points d'intérêt est réalisée par la fonction *ImageKeypoints* de Mathematica [6]. Cette fonction renvoie pour une image donnée, une liste de points d'intérêt déterminés par l'algorithme SURF.

4-3- Mise en correspondance des points d'intérêt

La mise en correspondance des points d'intérêt est réalisée par la fonction *ImageCorrespondingPoints* de Mathematica [6]. Cette fonction renvoie une liste de paires de points mis en correspondance sur les deux images, en mesurant la distance entre les descripteurs SURF des points des deux images.

4-4- Calcul de l'homographie

Cette étape de calcul se fait selon l'algorithme 2.2 et à partir des paires de points mis en correspondance. Ce calcul renvoie la matrice de l'homographie \mathbf{H} , ainsi que la liste des points classés statiques et vérifiant l'homographie. La valeur du seuil τ utilisée dans l'estimation de l'homographie par Ransac (Algorithme 2.2) est déterminée par expérimentation.

4-5- Détection des points en mouvement

Ce traitement se fait selon l'algorithme 2.4 et renvoie les points classés en mouvement sur une image par rapport à celle qui la précède. Les valeurs des seuils τ' et τ'' utilisés respectivement dans la 1^{ère} et la 2^{ème} étape d'élimination des faux points en mouvement sont déterminées par expérimentation.

4-6- Classification du mouvement

À partir des points classés en mouvement à l'étape précédente, il sera décidé de la nature du mouvement normal ou chute selon l'algorithme 2.5.

5- Images de synthèse : Traitement et interprétation des résultats

5-1- Génération des Images de synthèse

Afin de générer des images de synthèse animées par le logiciel POV-Ray, nous fixons certains paramètres: la position de la caméra, le point qu'elle regarde initialement, ainsi que l'angle de rotation, les sources de lumière (position, couleur, ombrage...), le nombre de trames de la séquence d'images, les coordonnées des éléments de la scène (Murs, plancher, cylindre mobile).

La figure (3.2.a) représente une vue latérale d'une scène intérieure à l'échelle 3cm:1m. Cette figure permet de calculer certains paramètres géométriquement :

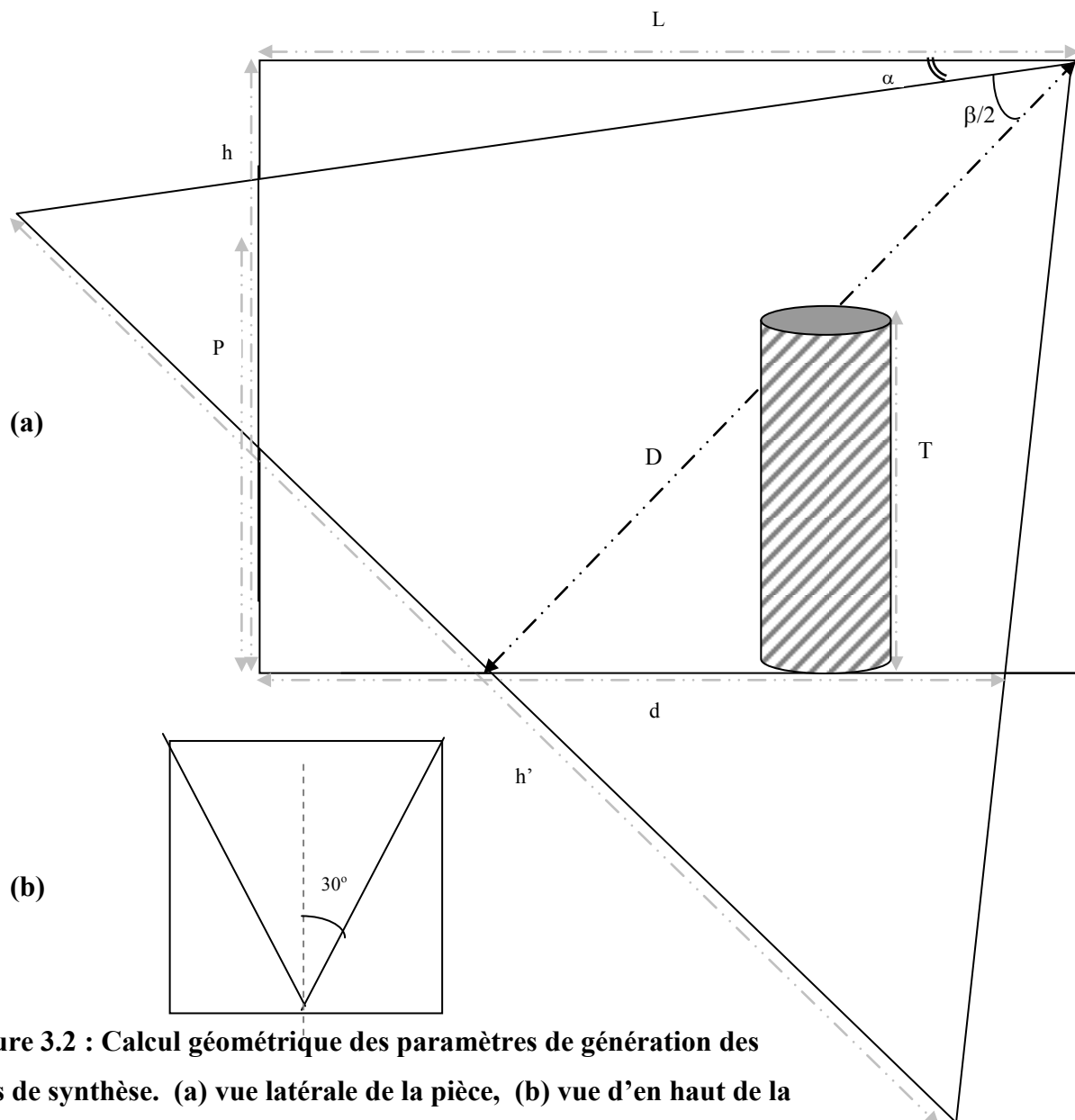


Figure 3.2 : Calcul géométrique des paramètres de génération des images de synthèse. (a) vue latérale de la pièce, (b) vue d'en haut de la

La pièce est de longueur $L=4\text{m}$ et de hauteur $h=3\text{m}$. La caméra est fixée au plafond à égale distance de deux murs qui sont en face l'un de l'autre (Figure 2.1 p. 31), soit à 3m du sol, à 1.50m des deux murs se faisant face et à 4m du mur qui lui fait face. Elle est inclinée vers le bas d'un angle α . La pièce a une porte de hauteur $P=2.50\text{m}$ et une fenêtre faisant face à la caméra. Le cylindre simulant une personne en déplacement a une hauteur $T=1.80\text{m}$.

Pour que la caméra puisse avoir un champ de vision couvrant le plancher sur une longueur $d=3.50\text{m}$ et le mur d'en face jusqu'au cadre supérieur de la porte (à $P= 2.13\text{m}$ du sol), son angle de vue β calculé géométriquement doit être de 70 degrés. Le calcul géométrique donne aussi un angle α de 12° , une valeur $D=4.16\text{m}$ représentant la médiatrice du cône de vision et une valeur $h'=5.73\text{m}$ représentant la longueur du plan qui sera projeté sur le plan image.

Ces différents paramètres ont été fixés de manière à respecter la proportionnalité des différents éléments de la scène ainsi que la dimension anthropomorphique du cylindre.

Une projection de ces valeurs sur le plan h' et par la suite sur le plan image permet de fournir à POV-Ray les coordonnées nécessaires pour générer la scène.

Dans la génération des séquences d'images de synthèse, nous faisons varier l'angle de rotation de la caméra à chaque expérimentation afin de déterminer la meilleure valeur qui convient à notre application. Compte tenu des dimensions de la pièce que l'on simule sur les images de synthèse, l'angle de balayage de la caméra va de -30° à $+30^\circ$. Cet intervalle est déterminé géométriquement (Figure 3.2.b).

Quant au déplacement du cylindre, nous l'effectuons à l'aide d'une liste de positions indiquant les coordonnées des différents points de la trajectoire du cylindre. Si on considère qu'une personne se déplace à une vitesse de 1m/s et que l'acquisition d'une vidéo se fait avec une vitesse de 30 images par seconde, le déplacement serait de $3\text{cm}=3\times 10^{-2}\text{m}$ par image. Si on considère les dimensions des images générées par POV_Ray (384×512), un déplacement normal sera de 2 pixels, ceci est obtenu par $(3\times 10^{-2} \times 384)/h'$.

La génération des images de synthèse se fait par l'algorithme suivant :

Début :

Fixer les paramètres de la caméra : position, point de fixation initial, angle de rotation θ .

Fixer les paramètres de la source de lumière : position, couleur, ombrage.

Générer le plancher, le plafond et les murs de la scène : paramètres d'un plan 2D, couleur, texture.

Générer l'objet en mouvement selon le modèle mathématique cylindre.

Définir la fonction spline pour le déplacement et la variation de la vitesse du mouvement.

Animer les objets en mouvement par une translation spline.

Fin.

Algorithme 3.1 : Génération des images de synthèse.

Les images synthétisées sont converties en images en niveau de gris pour leur traitement (Figure 3.3) :

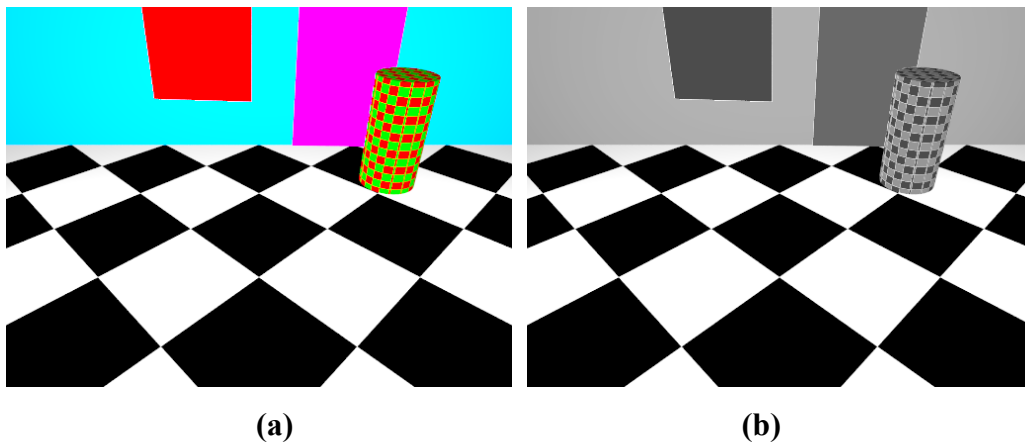


Figure 3.3: Image de la séquence vidéo synthétisée.

5-2- Extraction des points d'intérêt

Lors de cette étape et avant tout autre traitement, nous constituons, à partir des points d'intérêt extraits de chaque image, un ensemble de validation vérité terrain "un ground truth"

où les points en mouvement sont séparés des points statiques. Cet ensemble sera utilisé dans l'interprétation de nos résultats obtenus par l'algorithme de détection de mouvement (Algorithme 2.4). À cette fin, nous utilisons un masque obtenu par génération d'une image binaire à partir de l'image à niveaux de gris (Figure 3.4). Sur cette image binaire, seuls les pixels du cylindre en mouvement sont représentés en noir, le reste des pixels de l'image est en blanc.

Ainsi tout point extrait sur l'image à niveaux de gris devrait être classé en mouvement si sa position correspond à un pixel noir sur le masque, sinon il devrait être statique. C'est un moyen qui nous permet de valider les résultats de l'algorithme 2.4, lors de la détection de mouvement.

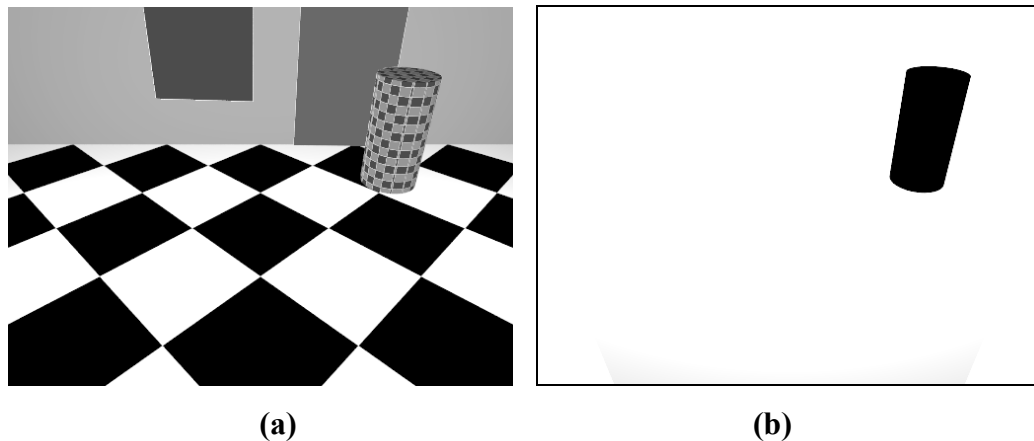


Figure 3.4 : Construction de l'ensemble de validation
(a) Image de synthèse (b) Masque binaire.

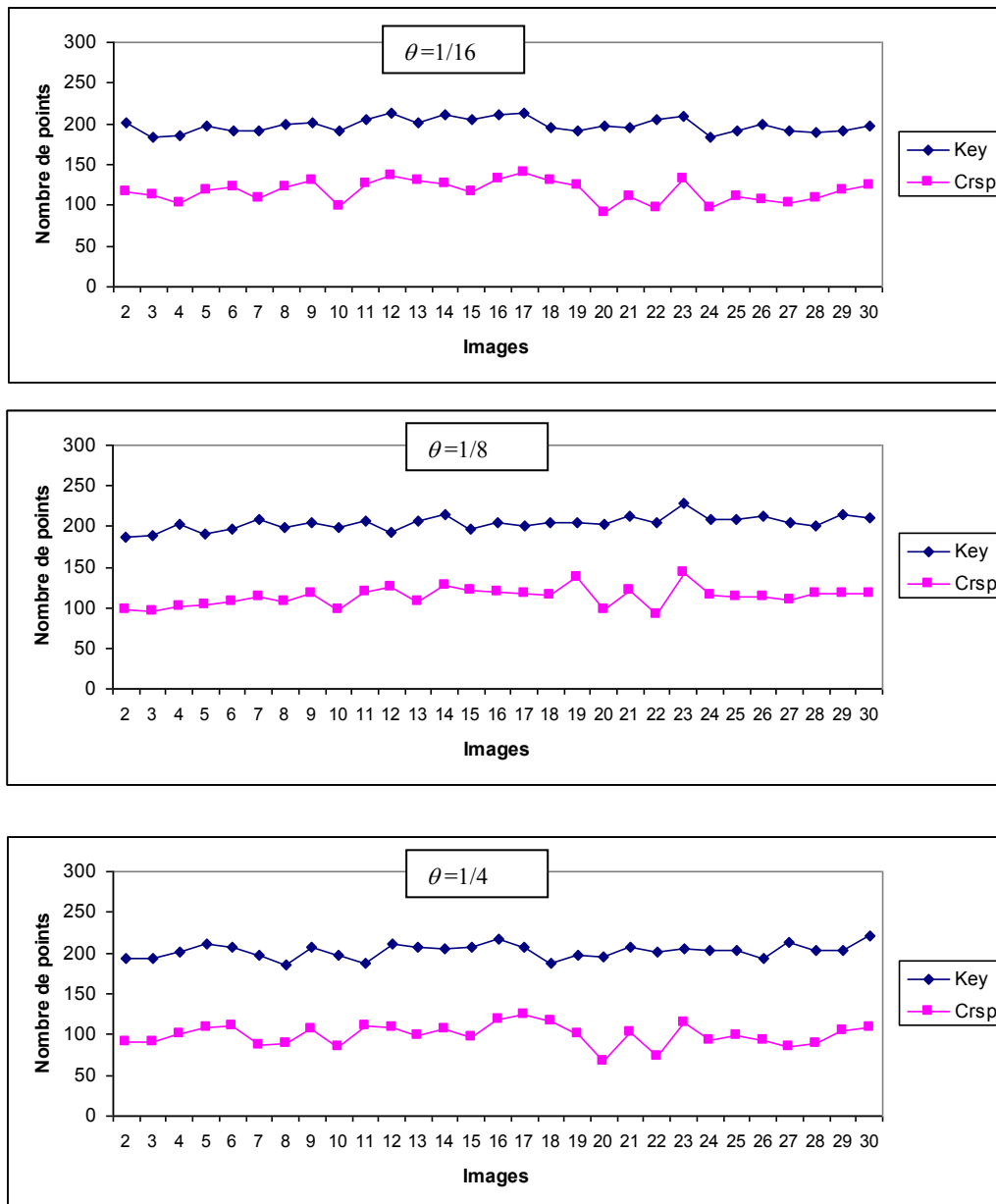
5-3- Mise en correspondance des points d'intérêt

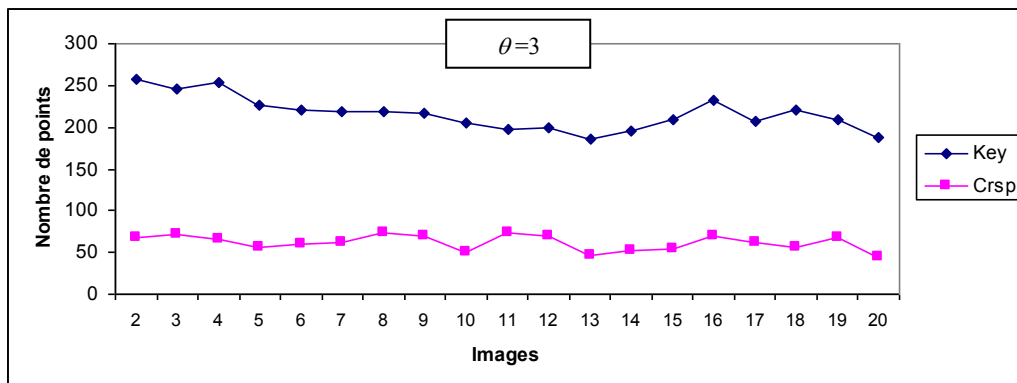
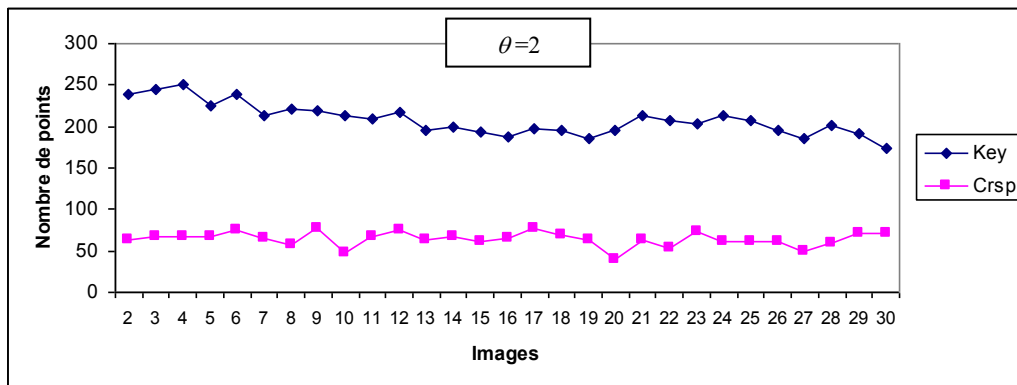
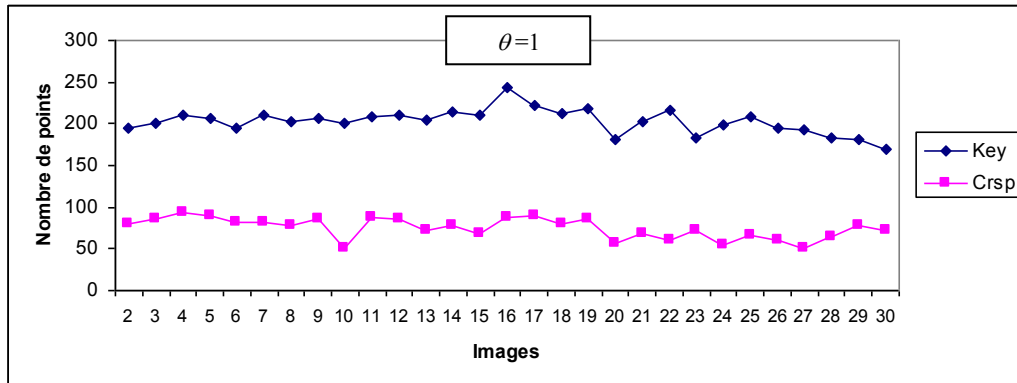
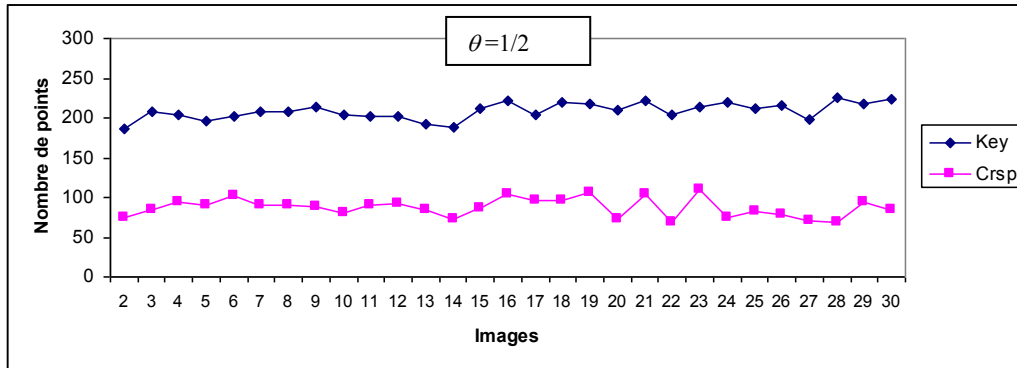
Les images de la séquence sont soumises deux à deux à la fonction *ImageCorrespondingPoints*, afin de mettre en correspondance les points d'intérêt de deux images successives.

Similairement à la phase d'extraction des points d'intérêt, pour chaque image à niveau de gris, nous utilisons le masque binaire lui correspondant afin de compter le nombre de points en mouvement et le nombre de points statiques parmi tous les points mis en correspondance. Cela permet de comparer les résultats de la mise en correspondance avec les résultats de

l'algorithme 2.4, en plus de comptabiliser les points d'intérêt qui ont été rejetés par la mise en correspondance.

L'extraction et la mise en correspondance des points d'intérêt de deux images successives de la séquence d'images, fournit les résultats suivants (Figure 3.5 pp. 69-71) pour différents angles de rotation θ (les valeurs de θ sont en degrés/image).





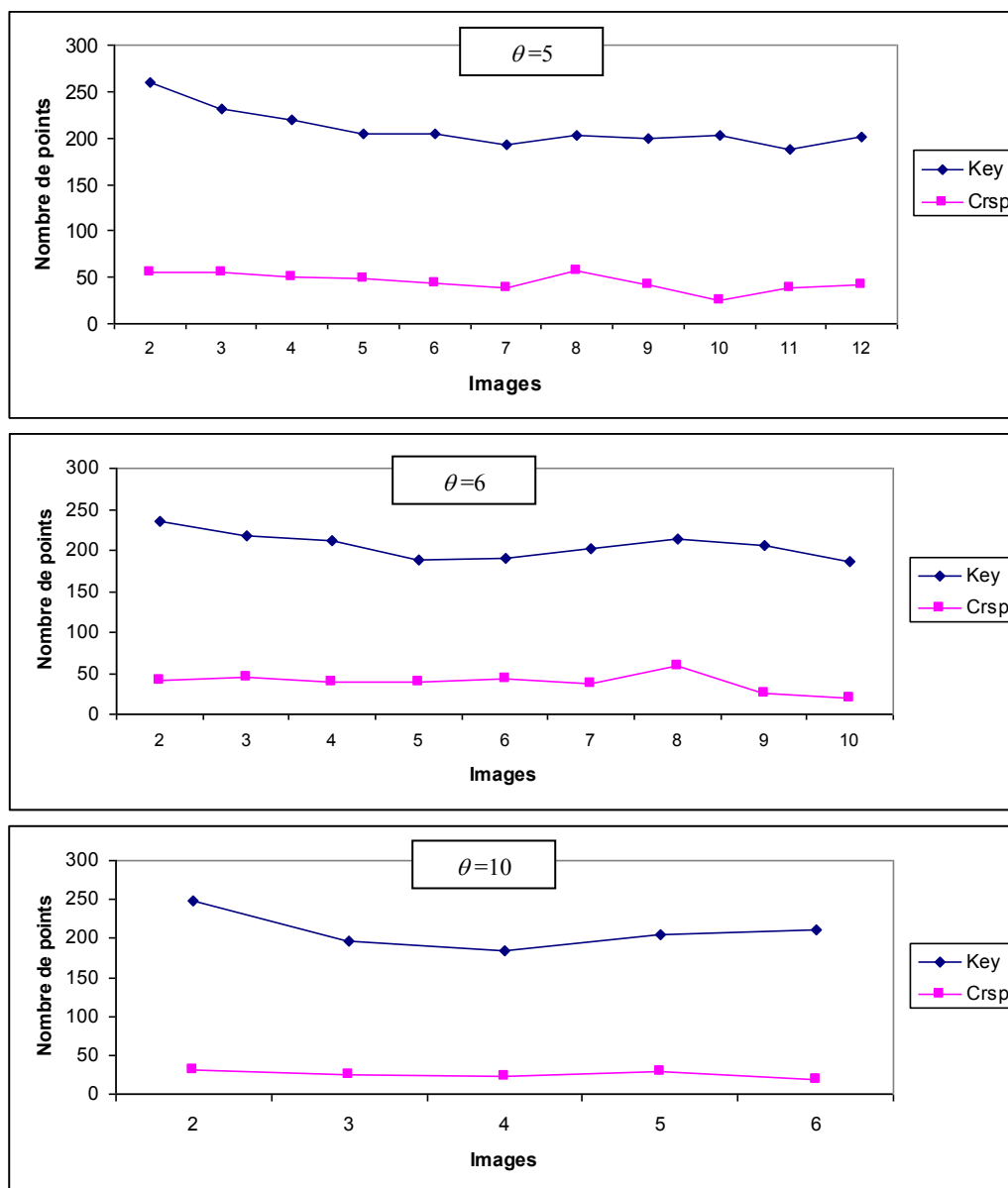


Figure 3.5 : Variation du Nombre de points d'intérêt (Key) et du nombre de points mis en correspondance (Crsp).

La figure 3.5 illustre la variation du nombre de points d'intérêt (Key) et du nombre de points mis en correspondance entre deux images successives, en fonction de l'angle de rotation θ . Un point de la courbe Key représente le nombre de points extraits sur une image et

un point de la courbe Crsp représente le nombre de points mis en correspondance entre cette même image et celle qui la précède.

La figure 3.5 montre que la courbe de points d'intérêt (Key) est relativement stable, par contre l'écart entre la courbe (Key) et la courbe (Crsp) est d'autant plus important que l'angle de rotation θ augmente. Donc le nombre de points mis en correspondance diminue lorsque θ augmente.

Cette interprétation se confirme avec la figure (3.6) qui montre la courbe de la moyenne du nombre de points d'intérêt n'ayant pas été mis en correspondance, en fonction de l'angle de rotation θ . Nous constatons que la courbe est croissante et le nombre de points non appariés augmente en fonction de l'angle de rotation θ .

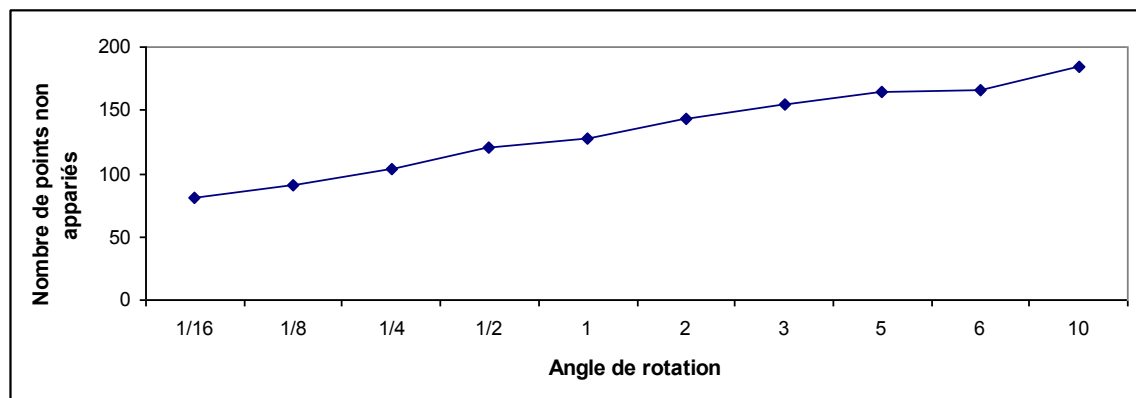
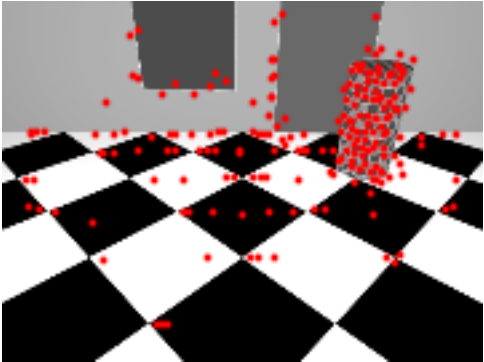
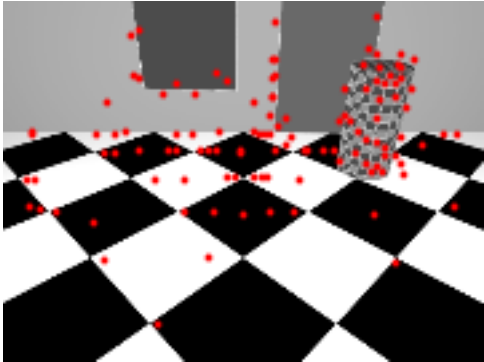
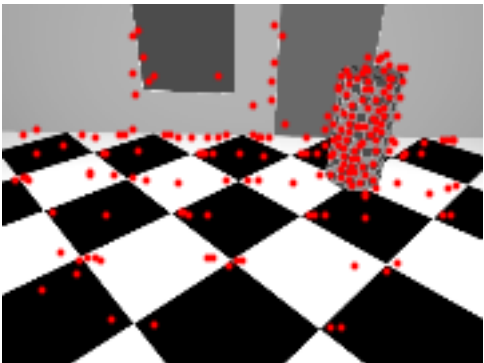
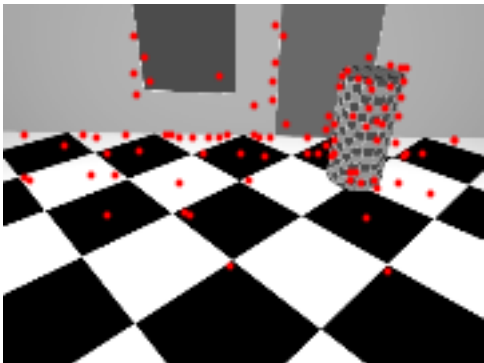
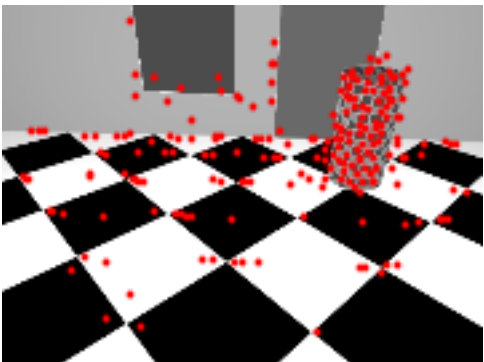
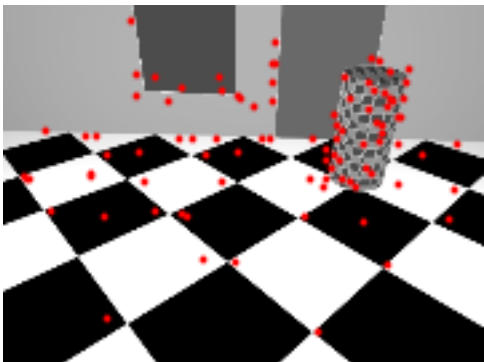


Figure 3.6 : Nombre de points d'intérêt non appariés en fonction de l'angle de rotation θ .

La figure 3.7 (pp. 73-74) suivante montre pour différentes valeurs de θ , une image de la scène avec les points d'intérêt extraits (colonne (a) de la figure) et la même image avec les points ayant une correspondance dans l'image qui la précède (colonne (b) de la figure). Nous constatons visuellement que les points mis en correspondance diminuent lorsque θ augmente. Cette figure (3.7) illustre donc, les résultats numériques présentés ci-dessus.

θ°	(a) Images avec les points d'intérêt	(b) Images avec les points mis en correspondance
$\frac{1}{16}$		
$\frac{1}{2}$		
1		

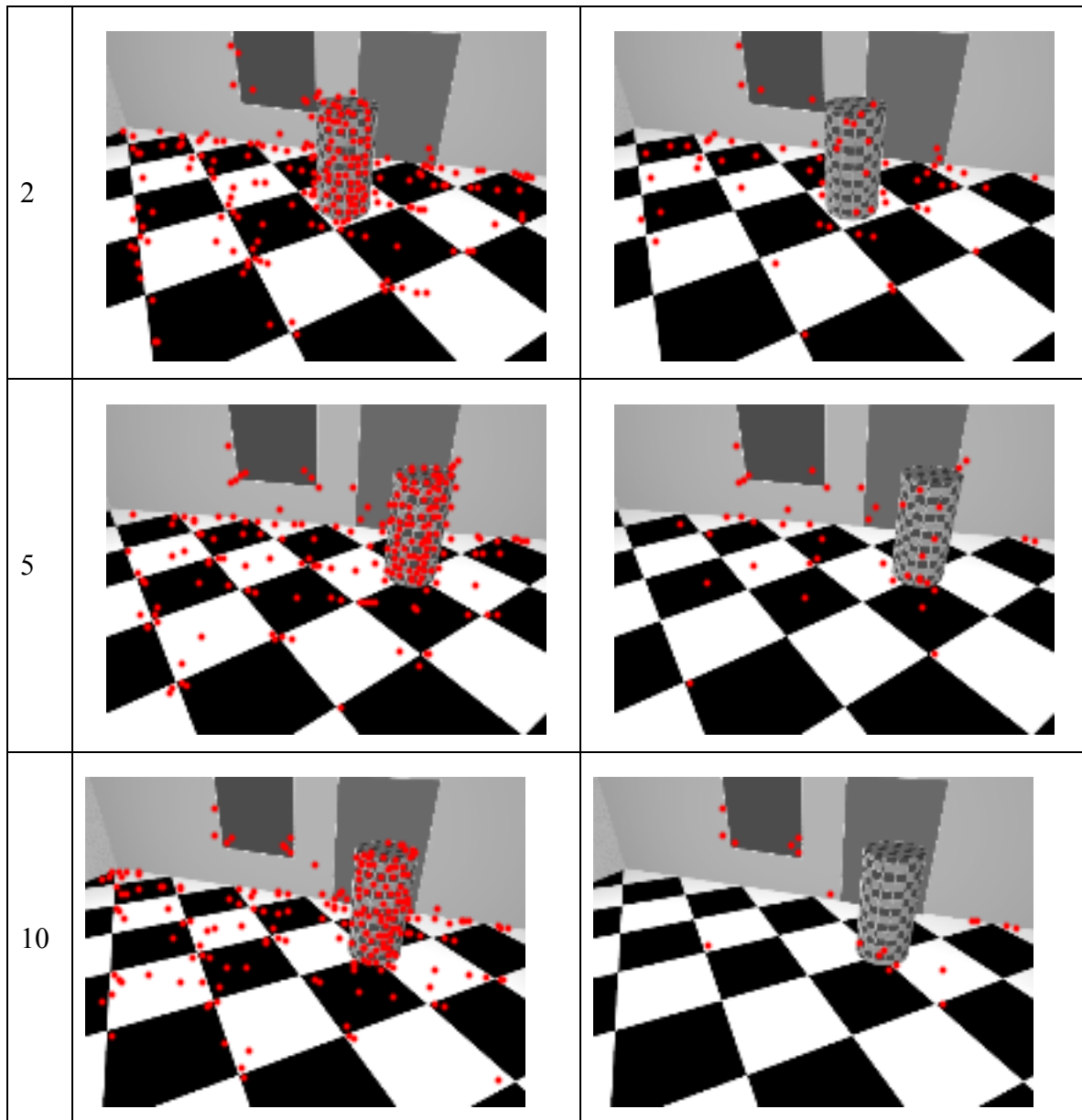


Figure 3.7 : Mise en correspondance de deux images successives.

(a) Images avec les points d'intérêt, (b) Images avec les points appariés.

La diminution des points mis en correspondance lorsque l'angle de rotation θ augmente, s'explique par le fait que l'arrière-plan change entre deux images successives si l'angle de rotation θ est relativement élevé. En effet, des objets de l'arrière-plan peuvent être occultés par l'objet mobile, de nouveaux objets apparaissent et disparaissent aux extrémités droite et gauche des images (exemple coin de la pièce).

5-4- Calcul de l'homographie

Les paires de points mis en correspondance sur deux images successives sont soumis à l'algorithme 2.2. Dans le cas des images de synthèse, la valeur du seuil τ utilisée dans l'estimation de l'homographie \mathbf{H} par l'algorithme 2.2 est égale à 1 pixel, cette faible valeur convient à la résolution relativement faible des images de synthèse que nous avons générées.

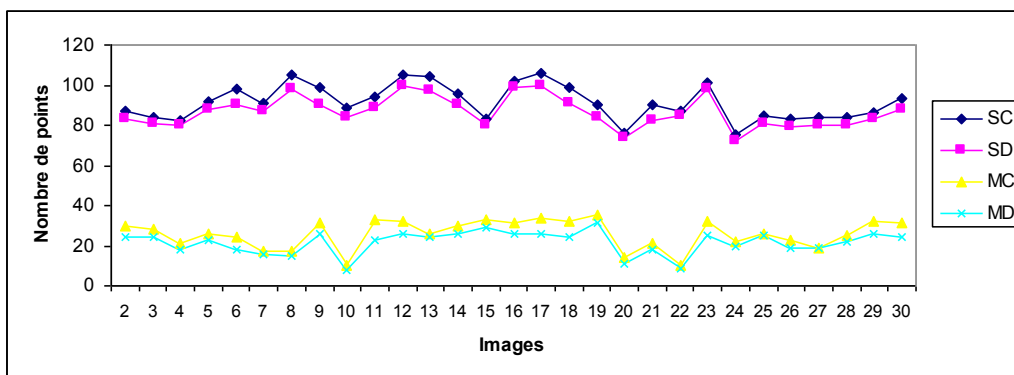
5-5- Détection des points en mouvement

Le traitement des images de synthèse par l'algorithme 2.4 se fait avec un seuil τ' relativement faible ($\tau' \in [0,1]$) compte tenu de la faible résolution des images et un seuil τ'' égal à 60 pixels. Pour déterminer la valeur de τ'' , nous avons positionné le cylindre loin de la caméra, au fond de la pièce contre le mur et nous avons calculé sa hauteur en pixels, puis nous l'avons placé à une position proche de la caméra de manière à ce qu'il reste entièrement dans le champ de vision de celle-ci et nous avons calculé sa hauteur en pixels. L'intervalle limité par ses deux valeurs nous a permis de faire plusieurs tests pour enfin choisir une valeur de 60 pixels.

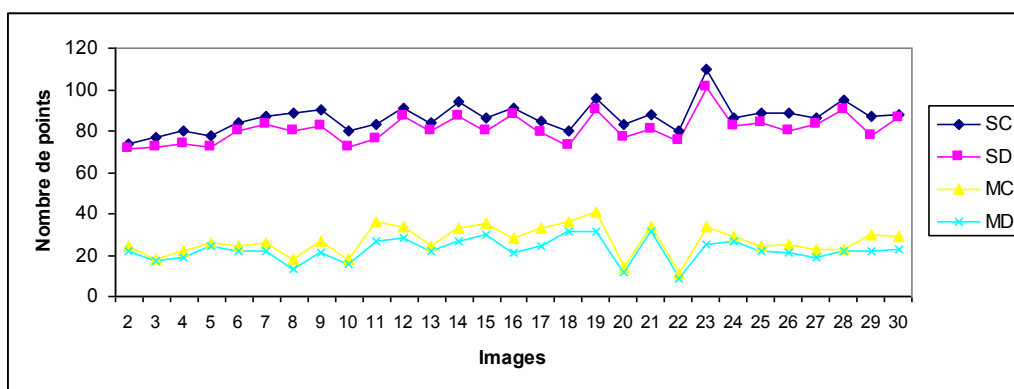
Nous interprétons dans ce qui suit, les résultats de détection de mouvement obtenus pour une séquence d'images que nous avons choisi parmi plusieurs. Dans cette séquence, le cylindre se déplace avec des vitesses variables. Cette variabilité de la vitesse nous permet de faire une étude de validité la plus complète qui soit.

La figure 3.8 (pp. 76-78) montre les résultats de l'algorithme de détection du mouvement pour différentes valeurs de l'angle de rotation θ :

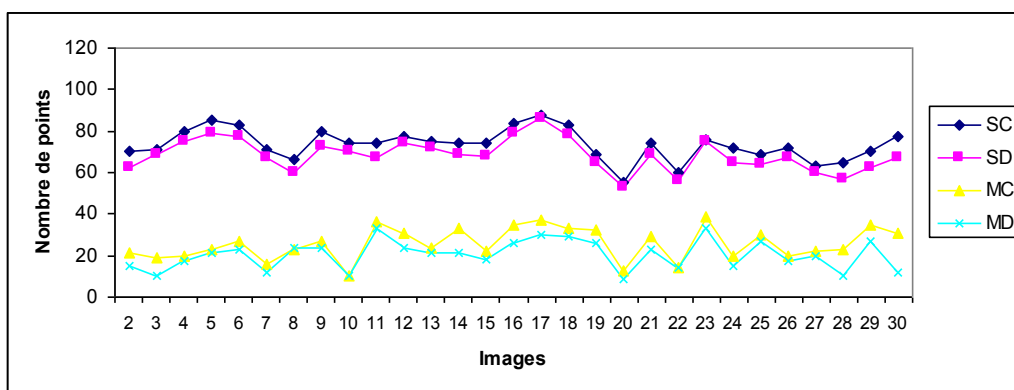
$$\theta=1/16$$



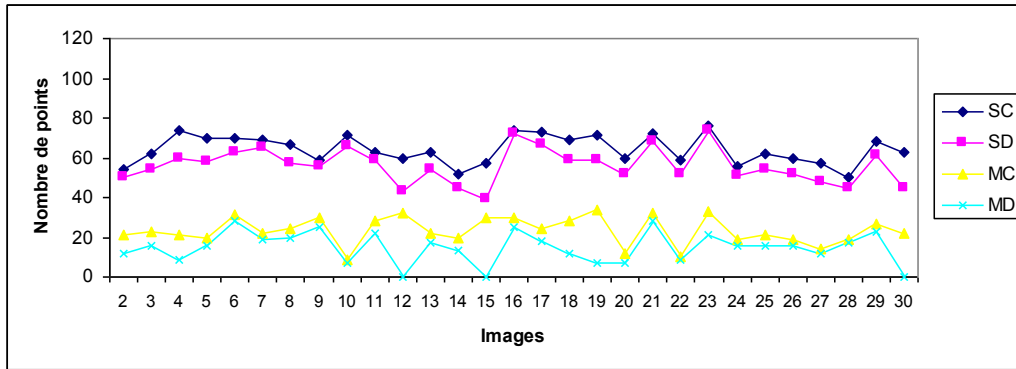
$$\theta=1/8$$



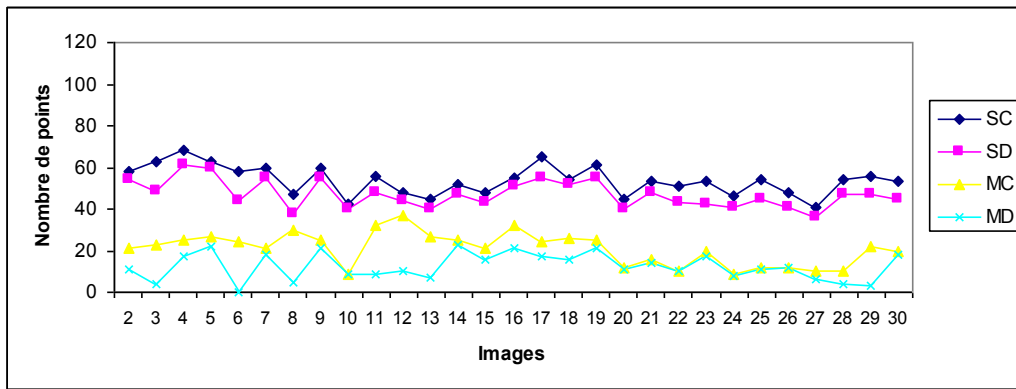
$$\theta=1/4$$



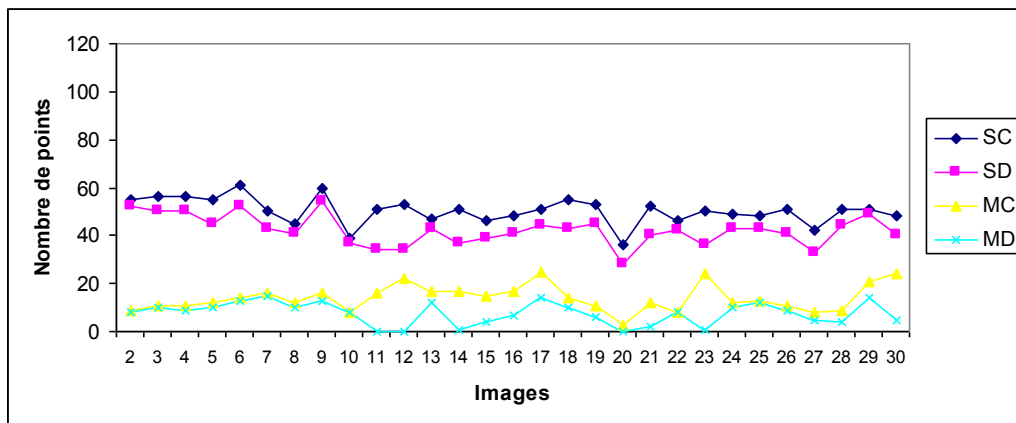
$$\theta = 1/2$$



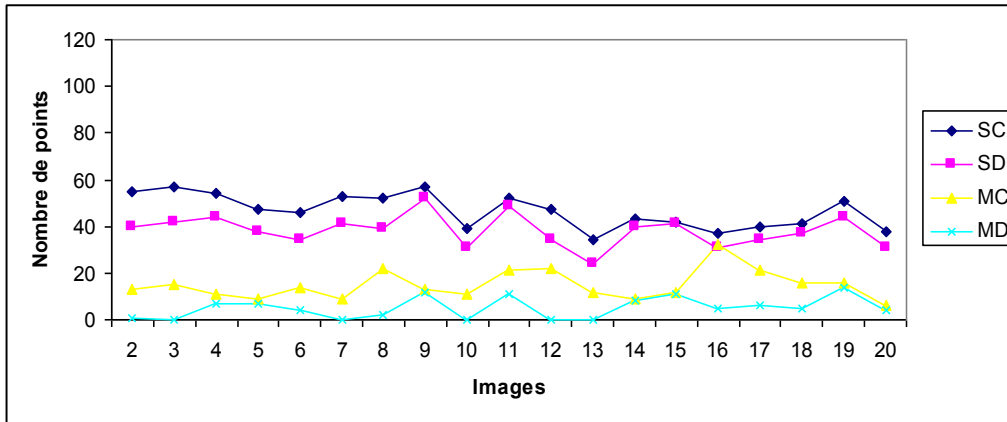
$$\theta = 1$$



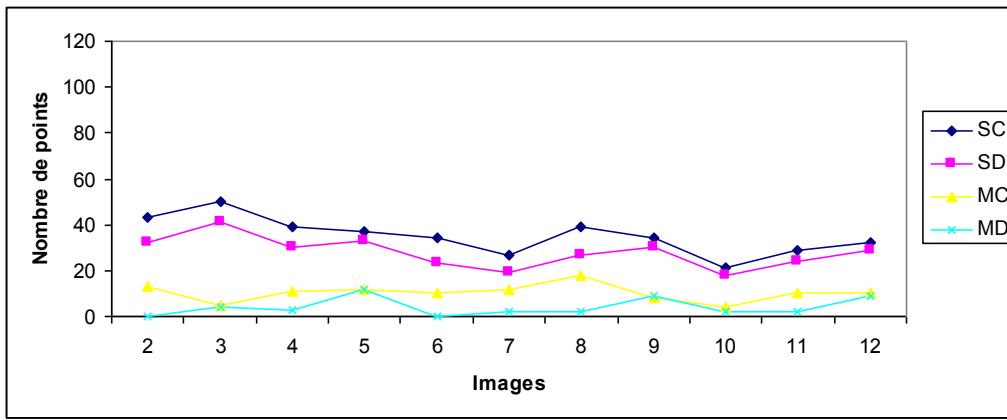
$$\theta = 2$$



$$\theta = 3$$



$$\theta = 5$$



$$\theta = 10$$

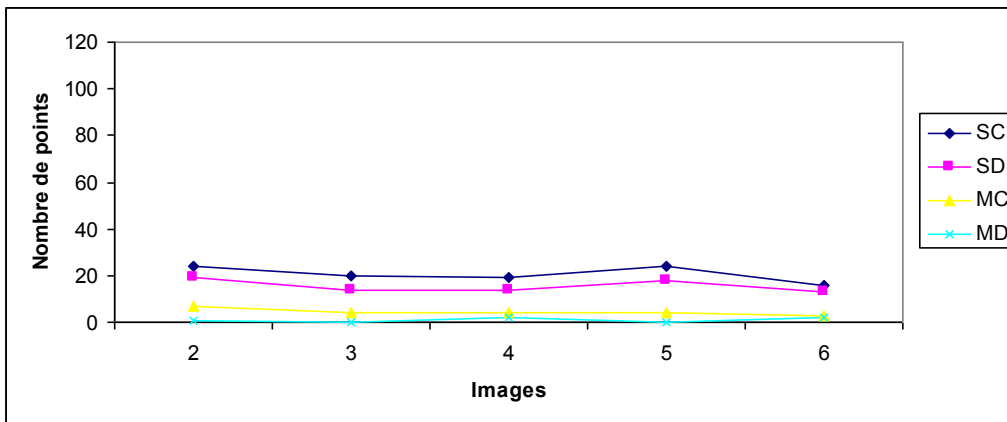


Figure 3.8 : Résultat de la détection du mouvement pour différentes valeurs de θ
 (SC) : Nombre de points statiques mis en correspondance (vérité terrain). (MC) : Nombre de points en mouvement mis en correspondance (vérité terrain). (SD) : Nombre de points statiques détectés (vrai négatifs). (MD) : Nombre de points en mouvement détectés (vrai positifs).

La figure 3.8 montre pour différentes valeurs de θ , la variation du nombre des points statiques détectés (SD) et la variation du nombre de points détectés en mouvement (MD) par l'algorithme 2.4, ainsi que la variation du nombre de points qui sont réellement en mouvement (MC) et la variation du nombre des points réellement statiques (SC) parmi les points mis en correspondance, selon la vérité terrain.

Cette figure, en plus de confirmer que le nombre de points mis en correspondance diminue lorsque θ augmente, elle montre que le nombre de points détectés en mouvement diminue également (courbe MD), mais d'une manière non linéaire par rapport au nombre de points réellement en mouvement (courbe MC). En effet l'écart entre les 2 courbes est de plus en plus réduit et l'algorithme dégénère à partir de θ égal à 5 degrés.

Nous pouvons confirmer cette constatation, par la figure 3.9. Sur cette figure et pour différentes valeurs de l'angle de rotation θ , la courbe MoyMD représente le nombre moyen dans la séquence d'images, de points détectés en mouvement par notre algorithme 2.4 (vrais positifs); la courbe MoyMC représente le nombre moyen dans la séquence d'images, de points réellement en mouvement (selon la vérité terrain) parmi les points mis en correspondance et MoyFP représente les moyennes des nombres de points classés faussement en mouvement (faux positifs), par l'algorithme 2.4, dans toute la séquence d'images.

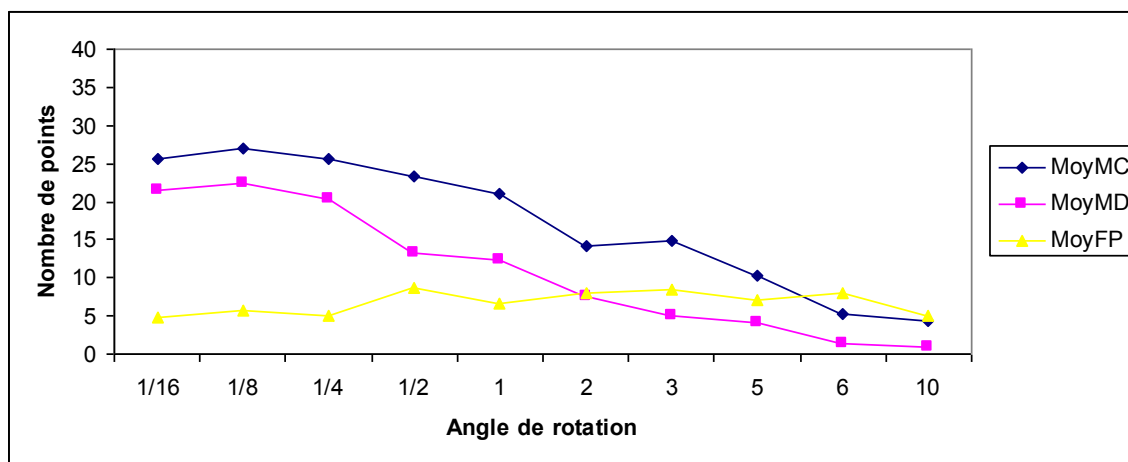


Figure 3.9 : Variation du nombre moyen de points détectés en fonction de θ

MoyMC : Vérité terrain; MoyMD : Vrai positifs; MoyFP : faux positifs

La figure 3.9 montre que pour des faibles valeurs de θ , l'écart entre la courbe des faux positifs (MoyFp) et les deux autres courbes est significatif. Cet écart tend à diminuer et les faux positifs deviennent plus nombreux que les vrais positifs à partir de θ égal à 3 degrés, car le nombre de points mis en correspondance a diminué. À partir de θ égal à 5 degrés, l'algorithme dégénère et toutes les courbes tendent vers 0.

La figure 3.10 montre le taux de points détectés en mouvement par l'algorithme 2.4 par rapport à l'ensemble de validation (vérité terrain).

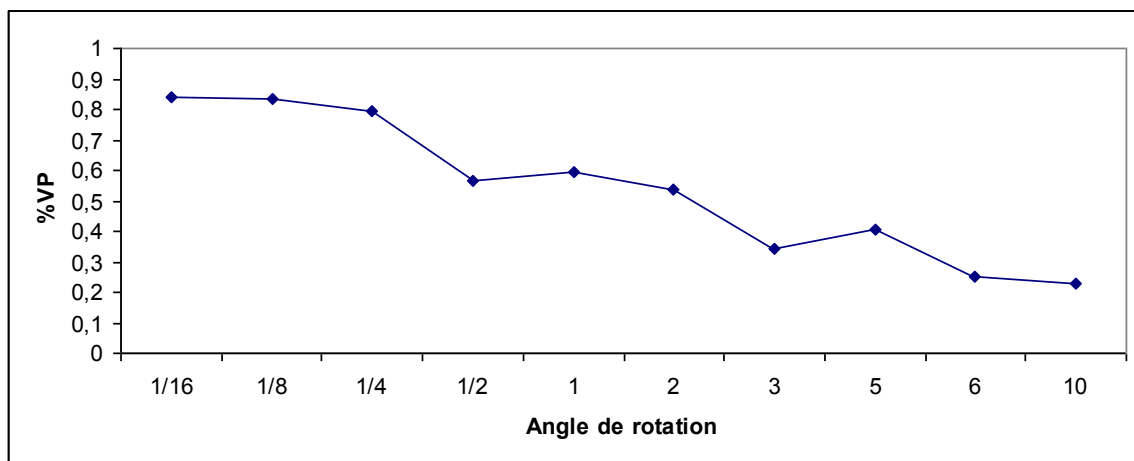


Figure 3.10 : Pourcentage de points détectés en mouvement en fonction de l'angle θ

$$\%VP = \text{MoyMD} / \text{MoyMC}$$

Pour de faibles valeurs de θ entre 1/16 et 1/4 degrés, le pourcentage des points détectés en mouvement est au dessus de 80 %, il diminue pour des valeurs de θ situées entre 1/2 et 3 degrés.

À partir de θ égal à 3 degrés, la courbe montre que ce pourcentage tend à augmenter, cela s'explique par le fait que le nombre de points mis en correspondance a diminué de manière significative. À partir de θ égal à 5 degrés, le pourcentage de points classés en mouvement diminue considérablement, car la mise en correspondance dégénère et l'algorithme 2.4 suit cette tendance et dégénère à son tour.

La figure 3.11 suivante (pp. 81-82), montre pour différentes valeurs de θ , une image de la scène avec les points mis en correspondance avec l'image qui lui précède (à gauche de la

figure) et la même image avec les points détectés en mouvement par notre algorithme 2.4 (sur la partie droite de la figure).

Nous constatons visuellement que le nombre des points classés en mouvement diminue lorsque θ augmente. L'algorithme 2.4 est donc validé pour θ inférieur à 5 degrés.

Cette figure (3.11) illustre les résultats numériques présentés dans cette section.

θ°	(a) Images avec les points mis en correspondance	(b) Images avec les points détectés en mouvement.
$\frac{1}{16}$		
$\frac{1}{8}$		
$\frac{1}{4}$		

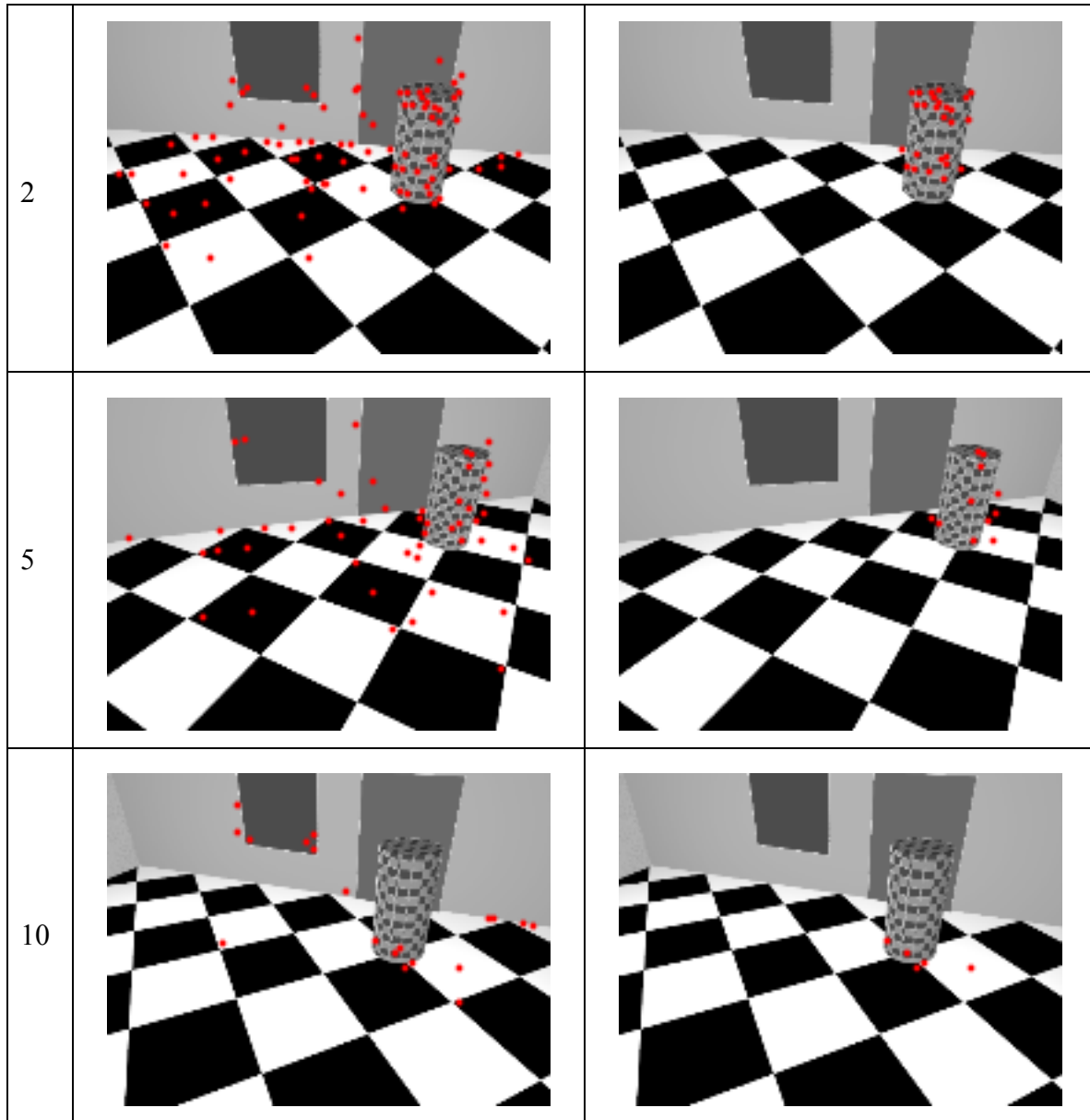


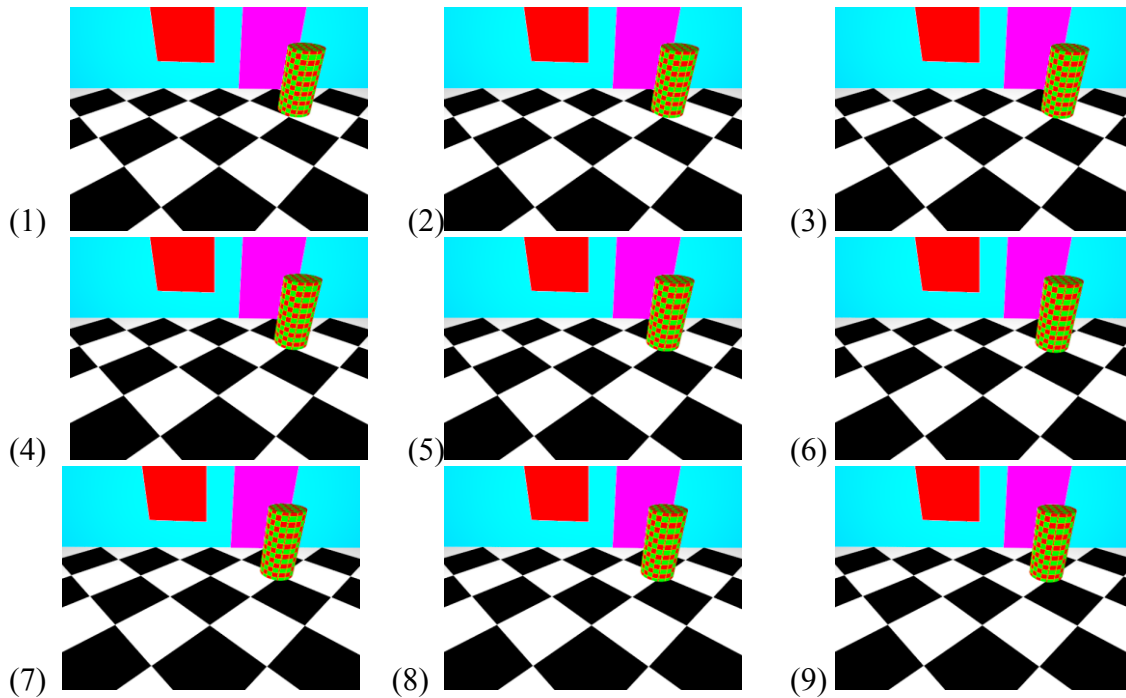
Figure 3.11 : Détection du mouvement. (a) Images avec les points appariés, (b) Images avec les points détectés en mouvement.

En résumé, l'algorithme 2.4 de détection de mouvement a montré sa validité ou de moins son utilité pour une valeur de l'angle de rotation de la caméra inférieure ou égale à 5 degrés. En effet, malgré la diminution du nombre de points détectés lorsque θ augmente, le mouvement sur l'image a été détecté. Cela ouvre de plus grandes possibilités à l'algorithme de détection de chutes.

5-6- Détection de chutes

Après avoir déterminé les points en mouvement sur l'image, ceux-ci seront classés en chute ou en mouvement normal selon l'algorithme 2.5. Dans cette partie du traitement, la valeur de l'angle de rotation θ n'intervient plus dans les calculs.

Nous avons généré des séquences d'images avec un mouvement à vitesse variable. Nous présentons dans ce qui suit les résultats obtenus à partir d'une séquence de 30 images (Figure 3.12 pp. 83-84) contenant du mouvement lent, légèrement brusque et brusque, correspondant à des translations des pixels entre deux images successives, appartenant à l'intervalle $[2, 15]$ pixels. Cette séquence représente un mouvement normal sur les 3 premières images permettant d'initialiser le processus de détection de chute^(*), puis un mouvement brusque simulant une chute sur les images 4 à 14, puis un mouvement normal sur les images 15 à 22, et enfin un mouvement brusque simulant une chute sur les images 23 à 30. Cela représente au total, 19 images simulant une chute et 10 images avec un mouvement normal, la première image étant exclue de la détection de mouvement.



^(*) Nous recherchons les points en mouvement sur Image₂ par rapport à Image₁, puis les points en mouvement sur Image₃ par rapport à Image₂, ensuite nous comparons la vitesse des points en mouvement sur Image₃ par rapport à Image₂, afin de classer le mouvement en normal ou chute. Cela signifie que la recherche du mouvement dans la séquence commence à partir de la deuxième image et la détection de chute commence à partir de la troisième image de la séquence.

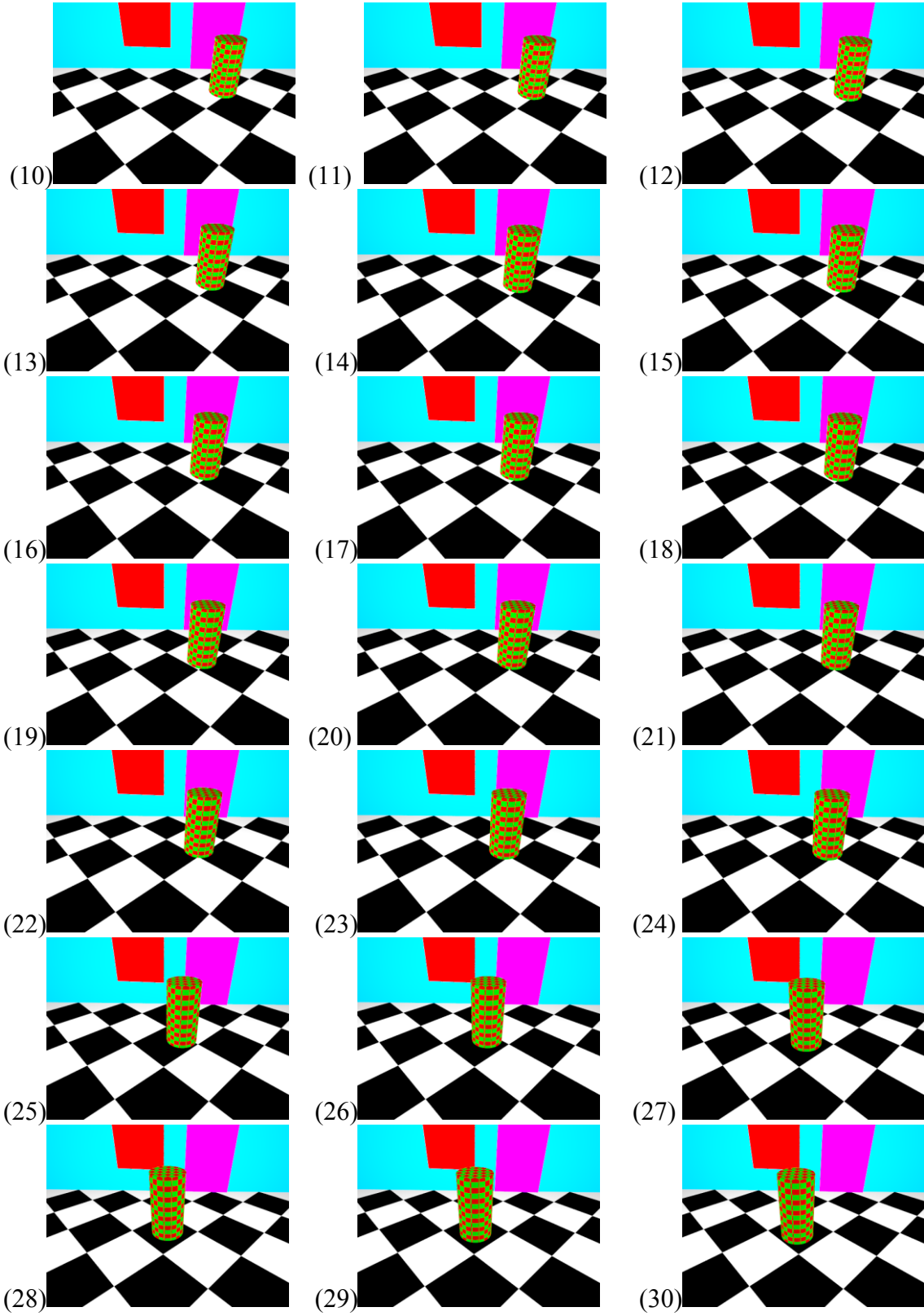


Figure 3.12 : Séquence de 30 images avec mouvement.

Rappelons que la classification du mouvement en chute se fait selon les inégalités (2.45) et (2.46) qui donnent: $d_{eucl}(n_1, n_2) > \rho = \frac{l}{a}$

où $d_{eucl}(n_1, n_2)$ est la distance euclidienne entre les points médians respectifs des nuages de points en mouvement sur deux images successives,

l est la hauteur de la boîte englobante et

a un paramètre dont la valeur est déterminée par expérimentation.

La figure 3.13 représente la variation du nombre de chutes détectées (c-à-d le nombre d'images avec chute) en fonction des valeurs de a .

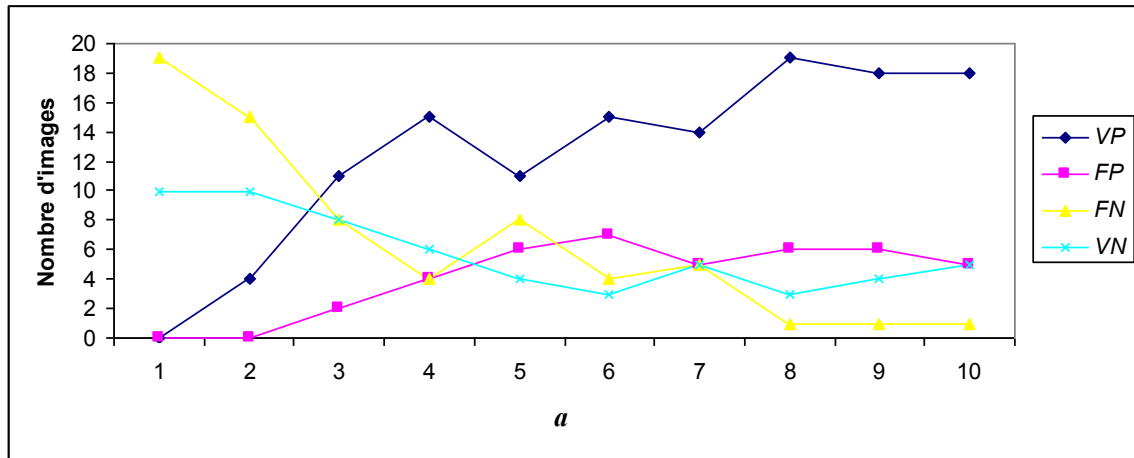


Figure 3.13 : Variation du nombre de chutes détectées.

VP : Vrai positifs, FP : faux positifs, VN : Vrai négatifs, FN : faux négatifs.

Sur la figure (3.13), la courbe (VP) représente le nombre d'images détectées avec chutes (vrai positifs), la courbe (FP) représente le nombre d'images ayant été classées en mouvement de chute alors que le mouvement y est normal (faux positifs), la courbe (VN) représente le nombre d'images avec un mouvement normal et qui sont classés en mouvement normal et la courbe (FN) représente le nombre d'images avec chutes qui n'ont pas été détectées.

La figure (3.13) montre aussi que pour $a \geq 6$, les courbes avec une classification fausse (FP) et (FN) ont tendance à décroître, alors les courbes représentant les classifications correctes (VP) et (VN) sont croissantes, particulièrement la courbe (VP) qui pour a égal à 10, montre que 18 chutes ont été détectées parmi 19, ce qui représente un résultat satisfaisant.

La figure (3.14) représente la variation de la F_{mesure} en fonction des valeurs de a . La F_{mesure} se calcule par :

$$F_{mesure} = \frac{2 \cdot \text{précision} \cdot \text{rappel}}{\text{précision} + \text{rappel}}$$

$$\text{où } \text{précision} = \frac{VP}{VP + FP} \quad \text{et } \text{rappel} = \frac{VP}{VP + FN}$$

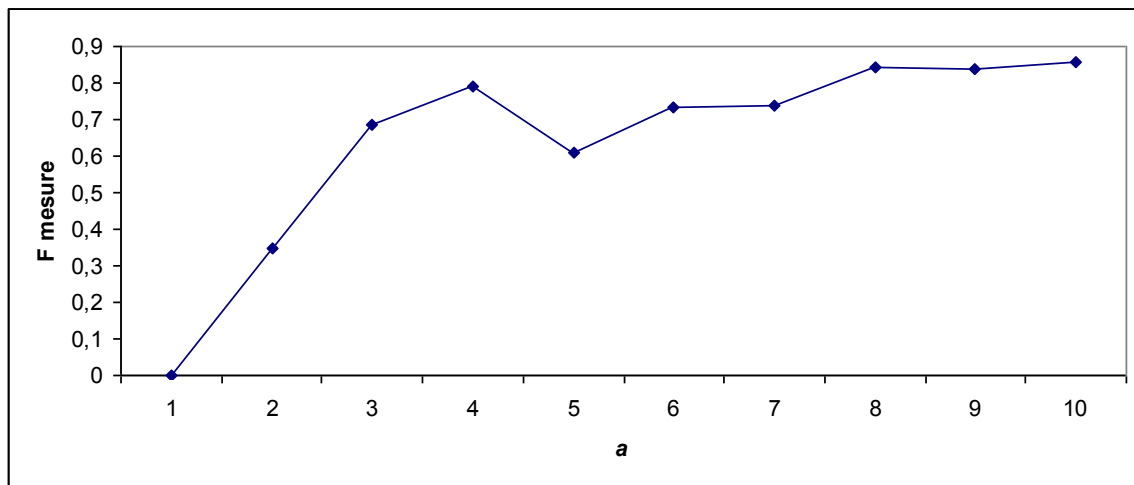


Figure 3.14 : Variation de la F_{mesure} en fonction des valeurs de a .

La figure (3.14) montre une courbe de la F_{mesure} qui atteint une valeur optimale autour de $a = 10$.

La figure (3.15) montre la variation du seuil ρ et de la distance des centroides n_1 et n_2 au cours de la séquence d'images pour une valeur de a égale à 10.

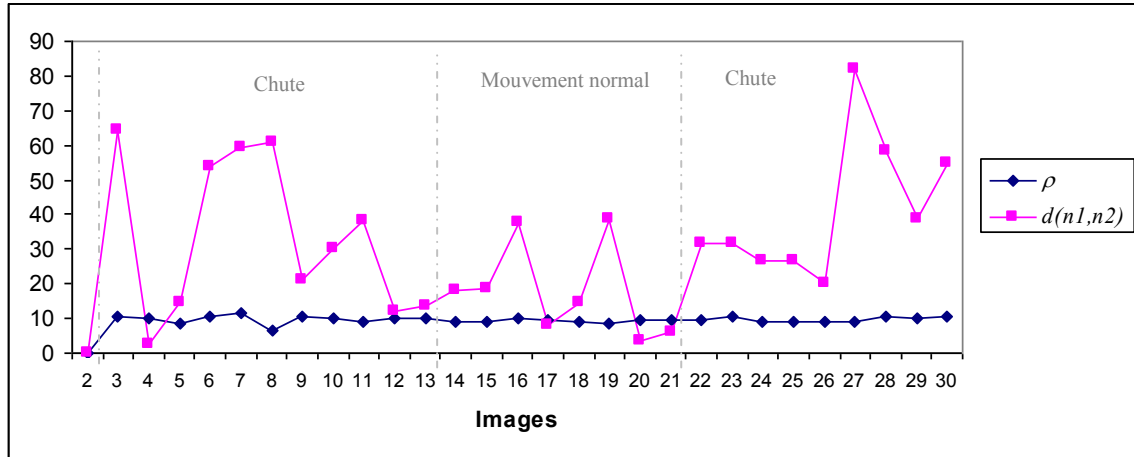
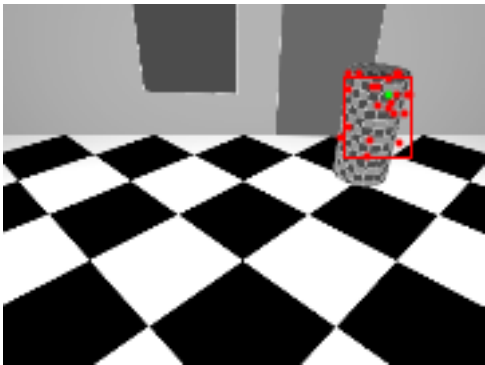
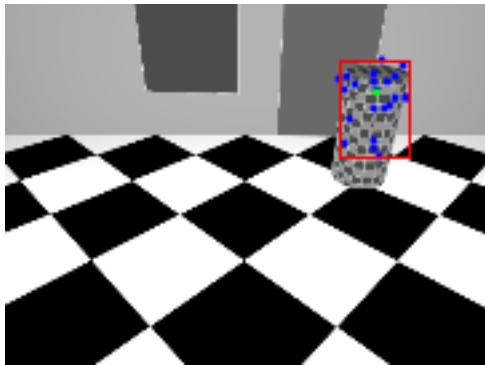
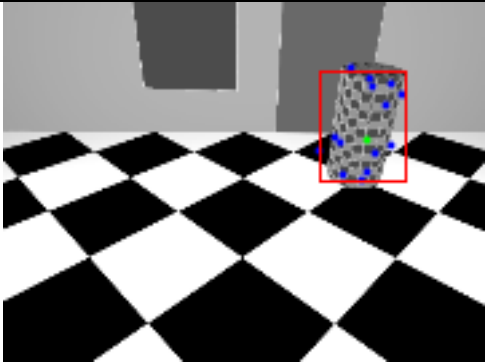
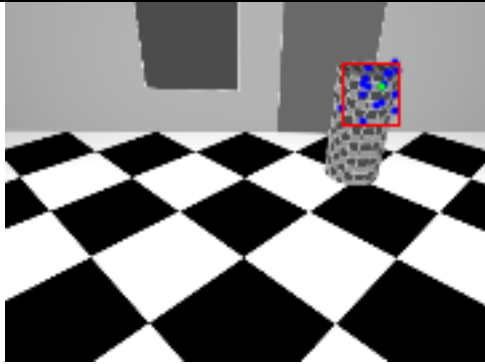
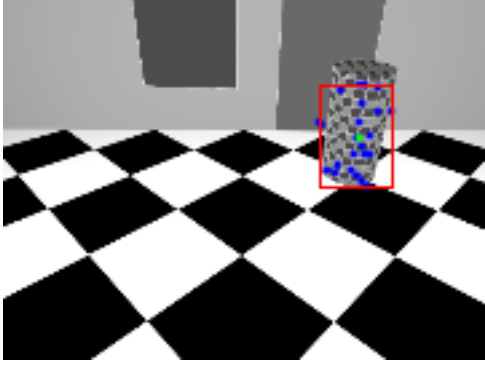
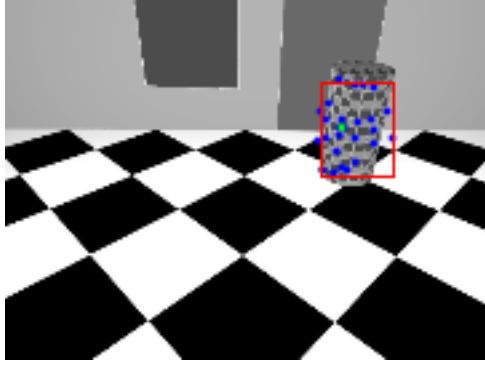


Figure 3.15 : Variation du seuil ρ et de la distance $d(n_1, n_2)$, pour $\alpha=10$.

La figure (3.15) montre que seule la chute sur l'image 4 n'a pas été détectée, alors que toutes les autres chutes ont été détectées. Nous remarquons aussi que pour les deux sous-séquences de chute, l'écart entre la courbe du seuil ρ et celle de la distance $d(n_1, n_2)$ est généralement plus significatif que pour la sous-séquence de mouvement normal. Cet écart s'explique par un déplacement brusque entre les deux images. Toutefois, dans la sous-séquence de mouvement normal, nous pouvons voir que l'écart entre les deux courbes présente deux pics. Cela est dû à un déplacement du nuage de points sur le cylindre qui n'est pas la conséquence d'un mouvement brusque, mais plutôt le résultat d'une sélection des points d'intérêt qui change en fonction de la position du cylindre par rapport à la source lumineuse. Ce phénomène engendre des fausses alertes (*FP*) dans la sous-séquence de mouvement normal. La figure (3.15) montre 5 fausses alertes, mais qui sont au plus deux fausses alertes successives. Afin de diminuer le cas de fausses alertes, nous proposons la procédure suivante :

Étant donné qu'un mouvement de chute ne peut se faire sur une image isolée, mais plutôt, sur plusieurs images successives, nous proposons d'analyser l'enchaînement des chutes sur la séquence d'images. Si les chutes sont détectées de façon isolée dans la séquence d'images, elles seront considérées comme des fausses alertes et aucune alerte réelle ne sera générée. Si elles constituent une chaîne de plusieurs images, elles seront prises en compte.

La figure (3.16) suivante (pp. 88-89) montre le résultat de la détection de chutes dans la sous-séquence des images [(4)-(14)] de la figure (3.12) :

Image	Résultat	Image	Résultat
4		5	
6		7	
8		9	

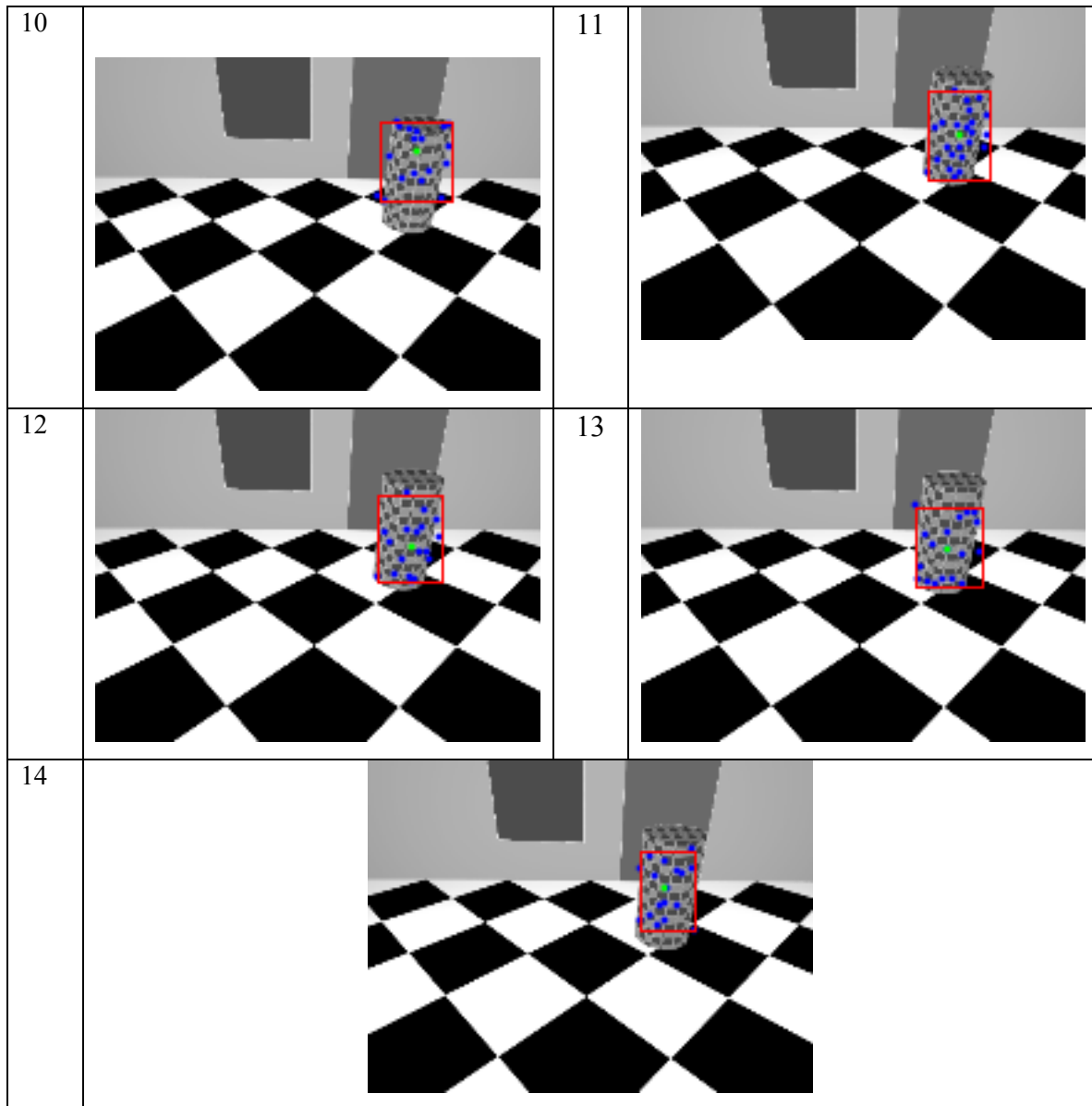


Figure 3.16 : Résultats du détecteur de chute (Algorithme 2.5).

La figure (3.16) représente les images 4 à 14 de la séquence traitée, sur chacune des images, des points en mouvement ont été détectés, parmi ces points nous avons gardé ceux dont les coordonnées sont dans l'intervalle du 5^{ème} percentile et du 95^{ème} percentile (les points en bleu), la boîte englobante (en rouge) est construite à partir des coordonnées maximale et minimale de ces points.

Le point médian du nuage de points est représenté en vert. C'est ce point qui permet de calculer le déplacement du nuage de points entre deux images successives.

La figure montre que toutes les chutes ont été détectées (points en bleu) à l'exception de l'image4 (points en rouge).

5-7- Conclusion sur le traitement des images de synthèse

L'expérimentation sur des images de synthèse nous a permis de tirer les conclusions suivantes :

Lorsque l'angle de rotation θ est faible et inférieur à 5 degrés, les résultats de la mise en correspondance et de la détection du mouvement sont meilleurs. Nous avons pu valider l'algorithme 2.4 pour une valeur de θ inférieure ou égale à 5.

Même si le nombre de points en mouvement est faible à cause du mouvement de la caméra ou à cause d'un mouvement rapide sur l'image ou de l'addition des deux mouvements, l'algorithme 2.4 arrive à détecter le mouvement sur l'image et préparer ainsi les données pour le détecteur de chutes.

L'angle θ n'est pas tenu d'être fixe pour toute la séquence d'images, puisque le calcul se fait entre une paire d'images successives.

Le détecteur de chute se base sur l'ensemble des points détectés en mouvement. C'est un classifieur qui dépend d'un seuil adaptatif qui tient compte de la taille de la boîte englobante. De ce fait il est moins sensible que s'il était fixe. Afin d'augmenter la robustesse du détecteur de chutes, nous avons proposé de rejeter les images où l'on détecte une chute de façon isolée et ce dans une sous-séquence d'images.

Notons que les images de synthèse utilisées ici sont de très basse résolution et que de meilleurs résultats de détection de points auraient vraisemblablement pu être obtenus pour des images plus réalistes et plus texturées. Nous verrons que sur des images réelles, nous obtenons un nombre de points SIFT plus important.

De même que le détecteur de mouvement, le détecteur de chute est validé pour les données de synthèse. Afin de tester la robustesse de cette méthode, nous présentons dans ce qui suit les résultats de l'expérimentation sur des images réelles.

6- Images réelles : Traitement et interprétation des résultats

Compte tenu des résultats de l'expérimentation sur des images de synthèse, nous avons situé les différents paramètres dans des intervalles de valeurs plus précis. Afin de tester la robustesse de notre méthode, nous soumettons les algorithmes à des images photographiques. Ce type d'images permet d'analyser la faisabilité de notre méthode sur des images nettes sans avoir à contrôler le phénomène de flou que l'on peut rencontrer sur des images vidéos lorsque des objets bougent rapidement (lorsque la vitesse d'obturation est insuffisante).

6-1- Détection de mouvement

La figure (3.17) (pp. 91-92) montre une séquence d'images où le robot (qui simule une personne âgée) se déplace à vitesse normale. Les images sont prises avec des angles de rotation variables, mais inférieurs à 5 degrés comme le suggèrent les résultats de l'expérimentation sur les images de synthèse.



(1)



(2)



(3)



(4)



(5)



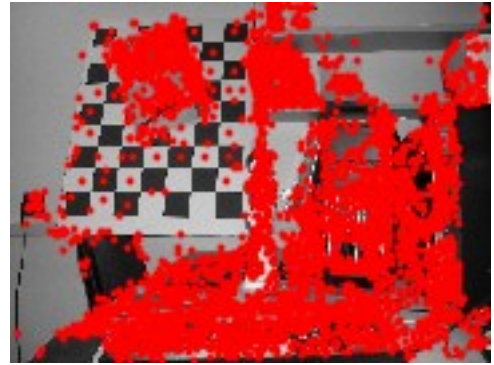
(6)

Figure 3.17 : Séquence d'images avec déplacement du robot et rotation de la caméra.

Pour ce type d'images les calculs ont été effectués avec les valeurs de seuils suivantes $\tau=2$, $\tau'=100$ et $\tau''=350$. Ces valeurs ont été déterminées par expérimentation. Elles sont plus élevées que pour le cas des images de synthèse. Ceci s'explique par la résolution des images qui est plus importante dans ce 2^{ème} cas.

La figure 3.18 montre les résultats de la détection du mouvement appliquée à l'image 3, choisie parmi la séquence d'images (l'image 3 est traitée par rapport à l'image qui la précède):

(a) Points de image3 ayant une correspondance sur image2.



(b) Points statiques de l'image 3 (algorithme 2.3)



(c) Points en mouvement sur l'image3 (algorithme 2.4)



Figure 3.18: Détection des points en mouvement.

Sur la figure 3.18 (b), sont affichés les points qui répondent à l'homographie entre l'image 3 et celle qui la précède, c'est-à-dire les points statiques appartenant à l'ensemble B obtenu par l'algorithme 2.3. Visuellement nous pouvons voir que le robot ne répond pas à l'homographie, car il est en mouvement entre les deux images. La partie (c) de la figure montre les points ayant été détectés en mouvement par l'algorithme 2.4.

Nous obtenons une bonne concentration de points sur le robot mobile. Donc l'algorithme 2.4 de détection de mouvement reste valide pour des images réelles.




6-2- Détection de Chute

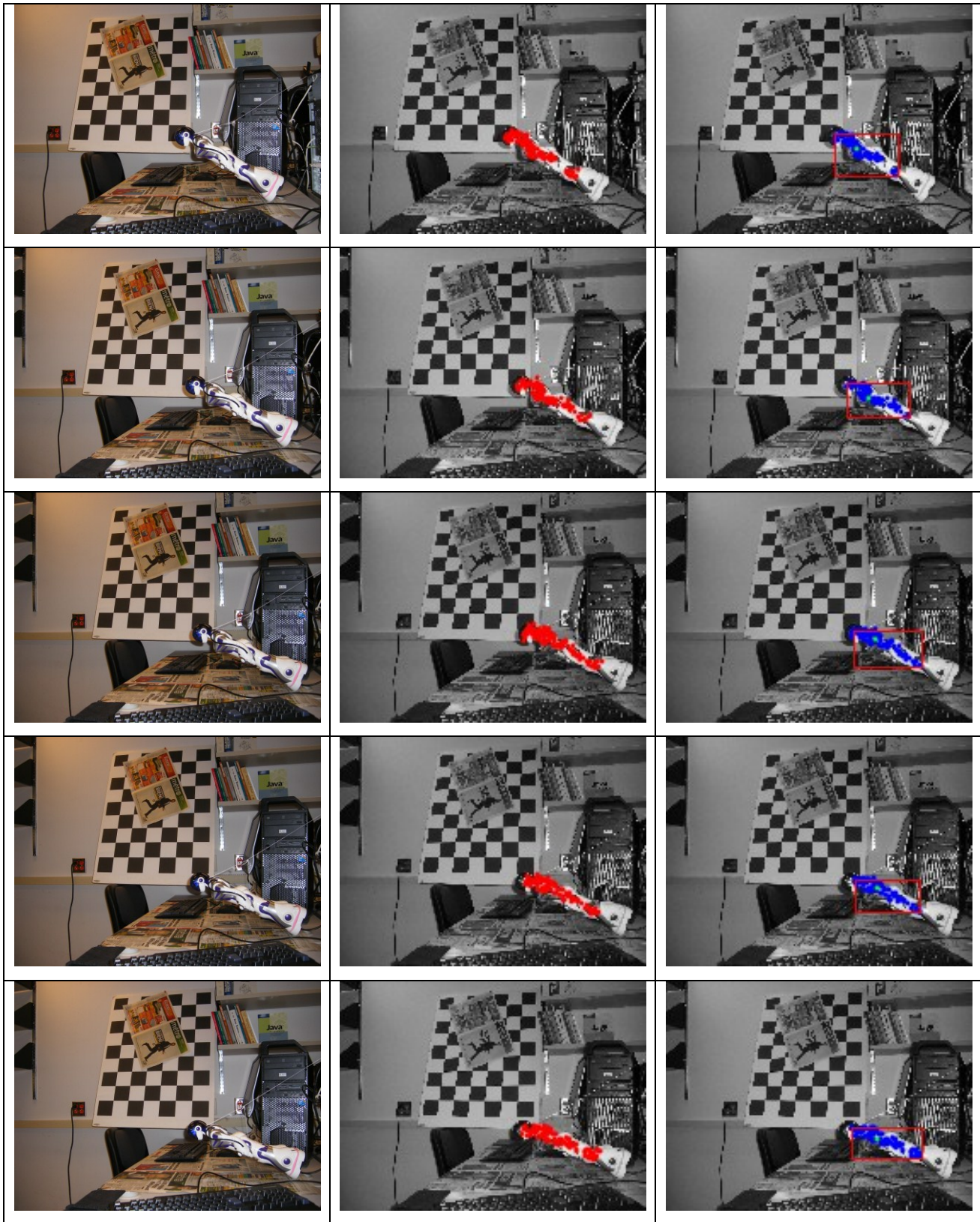
La figure 3.19 (pp. 94-96) montre dans la colonne (a) une séquence d'images où le robot est maintenu par une ficelle qui lui permet d'être à différentes positions de chute. Les images sont prises avec des angles de rotation variables, inférieurs à 5 degrés. Dans l'acquisition de cette séquence, la caméra effectue des rotations de droite à gauche et quelques fois de gauche à droite.

La colonne (b) de la figure 3.19 montre les points détectés en mouvement sur chaque image par l'algorithme 2.4.

La colonne (c) montre la classification du mouvement. Rappelons que la boîte englobante est utilisée pour le calcul de la valeur du seuil de classification ρ .

Les calculs ont été effectués avec les valeurs de seuils suivantes $\tau=2$, $\tau'=100$, $\tau''=350$ et $a=10$.

(a) Images	(b) Détection de mouvement par rapport à l'image précédente	(c) Détection de chute par rapport au mouvement précédent
		
		



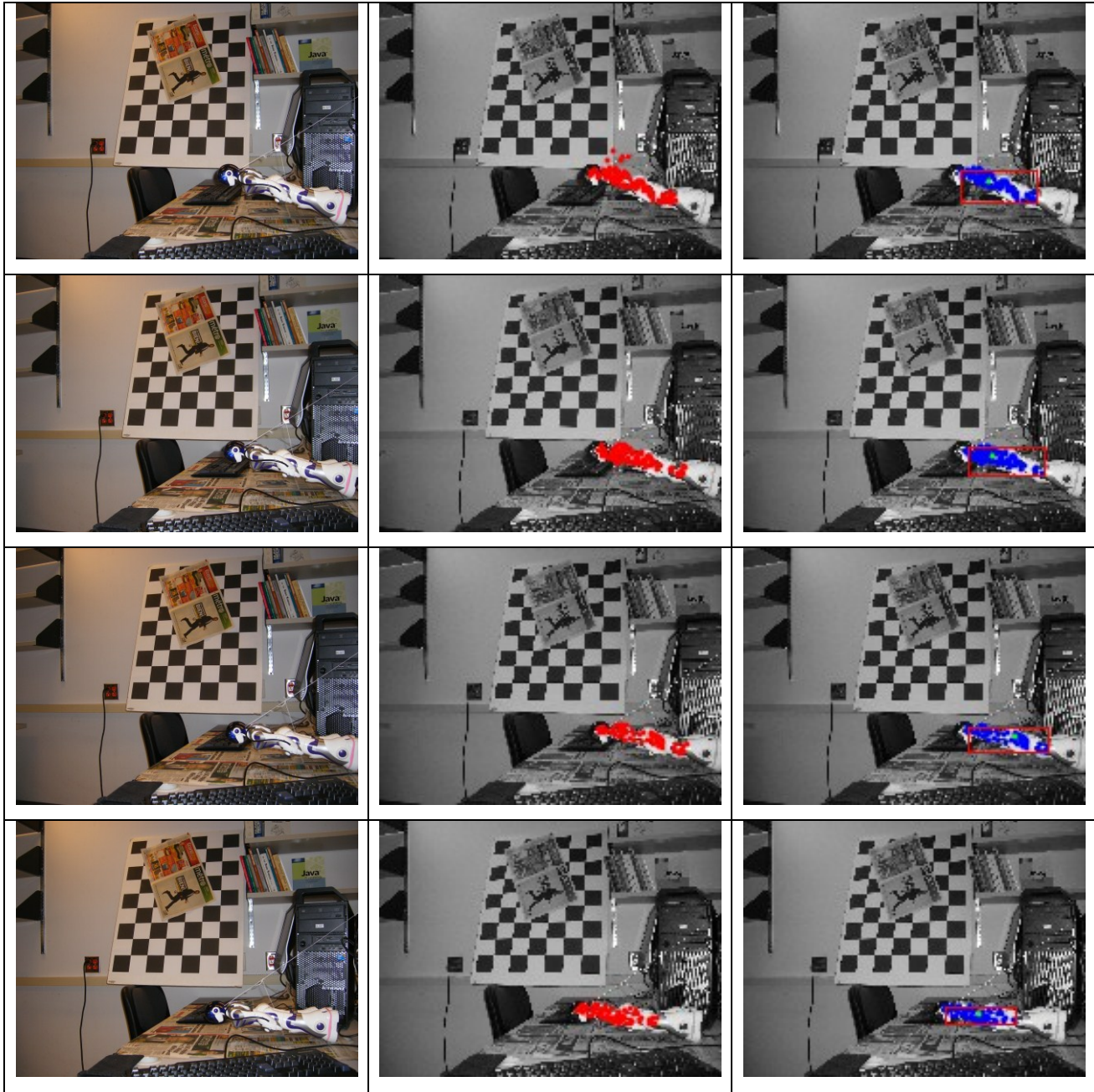


Figure 3.19 : Séquence de chute du robot.

(a) Images de la séquence, (b) Détection du mouvement, (c) Détection de chute.

Les résultats obtenus pour ces images photos sont satisfaisants, ils démontrent que notre méthode est valide et robuste pour des images réelles.

6-3- Discussion

Bien que les résultats précédents soient encourageants, nous avons été confrontés à d'autres résultats de qualité moindre, lors des différentes expérimentations que nous avons effectuées. Nous avons pu constater qu'il n'est pas évident d'appliquer cette méthode pour tout type de scène. Nous citons dans ce qui suit quelques exemples problématiques :

- Dans des scènes où il y a beaucoup de reflets (réflexions diffuses) sur l'arrière-plan, des points d'intérêt peuvent changer d'intensités lumineuses à cause de la rotation de la caméra et donc à cause du point de prise de vue par rapport à la position de la source de lumière.

- On peut aussi être confronté aux reflets sur le métal (réflexions spéculaires) qui peuvent induire du mouvement à travers la séquence d'images.

- Les images des scènes dont l'arrière-plan est beaucoup moins texturé que l'objet mobile, engendrent des homographies aberrantes, dans le sens où ces homographies sont calculées davantage à partir de points en mouvement que de points statiques.

- Si un mouvement est trop brusque entre deux images successives, il y a un risque qu'il ne soit pas détecté, si ces points en mouvement ne sont même pas appariés à cause de la rapidité du déplacement entre les deux images. Il serait donc judicieux dans le cas d'images vidéos d'avoir une vitesse d'acquisition des images vidéos adéquate afin de pallier à cet inconvénient.

- L'application sur des images vidéos, nécessite un temps d'obturation faible afin d'éviter d'avoir du flou sur les images lorsqu'un objet est en mouvement trop rapide.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

La détection de chutes par vidéosurveillance auprès des aînés ne cesse de mobiliser autour d'elle beaucoup de chercheurs en quête d'améliorer les résultats obtenus jusqu'à maintenant. Le travail que nous avons réalisé est une contribution modeste à l'automatisation de ce processus.

Dans le cadre de ce mémoire et après une étude bibliographique sur certaines méthodes se rapportant à la vidéosurveillance, nous avons présenté une approche de détection de mouvement et par suite de détection de chute, qui s'affranchit de la soustraction de l'arrière-plan d'une image dans le processus de détection du mouvement. Elle trouverait donc, son application pour des scènes où l'arrière-plan change d'une image à l'autre à cause du mouvement de la caméra. Notre approche se base essentiellement sur un mouvement de rotation pure de la caméra qui nous permet de calculer des homographies successives reliant deux paires d'images de la séquence.

Nous avons pu valider la méthode de détection de mouvement par calcul homographique, pour un angle de rotation de la caméra inférieur ou égal à 5 degrés/trame. La détection de chute est validée pour un seuil adaptatif proportionnel à la boîte englobante des points en mouvement. Cette méthode a fourni de bons résultats et des points en mouvement ont pu être détectés à travers la séquence d'images, ainsi que des mouvements brusques que nous assimilons à des chutes, compte tenu du contexte de l'application.

Cependant, l'utilisation de la méthode Ransac dans l'affinement de l'ensemble des points détectés en mouvement, nécessite un choix rigoureux du seuil. Dans ce présent travail, nous l'avons déterminé par expérimentation. Nous proposons d'utiliser dans des travaux futurs un seuil adaptatif qui tient compte de la dispersion des points du nuage de points, dans la détection de mouvement, comme nous l'avons fait dans la détection de chutes.

L'expérimentation sur des images de scènes réelles a aussi montré que cette méthode fonctionne pour des scènes dont l'arrière-plan est plus texturé que l'entité mobile et que les surfaces réfléchissantes sont problématiques dans la détection du mouvement.

L'application de cette méthode sur des images vidéos présentant différents problèmes habituels tels que la forte compression, le flou provoqué par des mouvements rapides, la présence de reflets lumineux sur certaines surfaces..., constituerait un défi pour la robustesse de cette méthode, qu'il n'est pas impossible de relever. Nous avons fait l'expérience sur des images vidéos et les résultats sont prometteurs, il s'agirait de déterminer adéquatement certains paramètres comme la bonne vitesse d'obturation pour avoir une meilleure qualité d'images et donc une meilleure extraction des points d'intérêts, comme nous avons pu le démontrer à travers les images photos.

RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] http://www.boschsecurity.fr/content/language1/downloads/Guide_de_la_videosurveillance_-_FFPP-final.pdf. (Guide de la vidéosurveillance 2010. Fédération française des Ports de Plaisance).
- [2] <http://www.cvg.rdg.ac.uk/slides/pets.html>
PETS: Performance Evaluation of Tracking and Surveillance.
- [3] <http://www.povray.org/>
- [4] www.sos-websolutions.com/portfolio/internetcameraiporan.php
- [5] "Vieillesse de la population, état fonctionnel des personnes âgées et besoins futurs en soins de longue durée au Québec". Institut National de Santé Publique du Québec, Mars 2010.
- [6] <http://www.wolfram.com/mathematica>
- [7] M. Alwan, P.J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe et R. Felder. "A Smart and Passive Floor-Vibration Based Fall Detector for Elderly". *Proceeding of the 2nd IEEE International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA'06)*, vol. 1, pp. 1003-1007, Apr. 2006.
- [8] D. Anderson, J. Keller, M. Skubic, X. Chen et Z. He. "Recognizing Falls from Silhouettes". *Proceedings of the 28th Conference of the IEEE. EMBS Annual International Conference*, pp. 6388-6391, New York City, USA, Aug. 2006.
- [9] E. Andrade, S. Blunsden et R. Fisher. "Performance Analysis of Event Detection Models in Crowded Scenes". *Proceedings of Workshop on Towards Robust Visual Surveillance Techniques and Systems, at Visual Information Engineering*, pp. 427-432. Bangalore, India, Sep. 2006.

- [10] E. Auvinet, E. Grossmann, C. Rougier, M. Dahmane et J. Meunier. "Left-Luggage Detection using Homographies and Simple Heuristics". *Proceedings of 9th IEEE International Workshop on PETS*, pp. 51-58, New York, Jun. 2006.
- [11] E. Auvinet, L. Reveret, A. St-Arnaud, J. Rousseau et J. Meunier. "Fall Detection Using Multiple Cameras". *30th Annual International IEEE, EMBS Conference*, pp. 2554-2557. Vancouver, British Columbia, Canada, Aug. 2008.
- [12] J.L. Barron, D. Fleet et S. Beauchemin. "Performance of Optical Flow Techniques". *International Journal of Computer Vision*, vol. 12, no 1, pp. 43-77, 1994.
- [13] H. Bay, T. Tuytelaars et L. Van Gool. "SURF: Speeded Up Robust Features". *Proceeding of the 9th European Conference on Computer Vision, ECCV 2006*, pp. 404-417, Graz, Autriche, May 2006.
- [14] H. Bay, A. Ess, T. Tuytelaars et L. Van Gool. "SURF: Speeded Up Robust Features". *Computer Vision and Image Understanding*, vol. 110, no 3, pp. 346-359, 2008.
- [15] S. Brutzer, B. Hoferlin et G. Heidemann. "Evaluation of Background Subtraction Techniques for Video Surveillance". *IEEE Computer Vision and Pattern Recognition Conference*, pp. 1937-1944, Jun. 2011.
- [16] J. Canny. "A Computational Approach to Edge Detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no 6, pp. 679-714, Nov. 1986.
- [17] M. Chan, E. Campo et D. Estève. "PROSAFE, A Multisensory Remote Monitoring System For the Elderly or the Handicapped in Independent Living for Persons with Disabilities and Elderly People". *Proceedings of The 1st International Conference on Smart Homes and Health Telematics*, pp. 89-95, Sep. 2003.

- [18] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt et L. Wixson. "A System for Video Surveillance and Monitoring". Final Report, CMU-RI-TR-00-12, Robotics Institute, CMU, May 2000.
- [19] R. Cucchiara, C. Grana, M. Piccardi et A. Prati. "Detecting Moving Objects, Ghosts, and Shadows in Video Streams". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1337-1342, Oct. 2003.
- [20] F. Cupillard, A. Avanzi, F. Bremond, M. Thonnat. "Video Understanding for Metro Surveillance". *IEEE International Conference on Networking, Sensing and Control*, vol. 1, pp. 186-191, Mar. 2004.
- [21] M. Dahmane. "Système de Vidéosurveillance et de Monitoring". *Mémoire de maîtrise en informatique*, Université de Montréal, Oct. 2004.
- [22] M. Dahmane, J. Meunier. "Real-time Video Surveillance with Self Organizing Maps ". *Proceedings of the IEEE 2nd Computer and Robot Vision Conference*, pp.136-143, 2005.
- [23] M. Dahmane. "Analyse de Mouvements Faciaux à Partir d'Images Vidéo". Thèse de Ph.D, Université de Montréal, Dec. 2011.
- [24] E. Dubrofsky. "Homography Estimation". *A Master's Essay. Faculty of Graduate Studies*. The University of British Columbia. Mar. 2009.
- [25] M. Fischler et R. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". *Communications of the ACM*, vol. 24, pp. 381–395, Jun. 1981.
- [26] D. Forsyth et J. Ponce. "Computer Vision, A Modern Approach". *Prentice Hall*, 2003.

- [27] L. Fuentes et Sergio A. Velastin. "People Tracking in Surveillance Applications". *Image and Vision Computing*, vol. 24 pp. 1165–1171, 2006.
- [28] F. Fusier, V. Valentin, F. Bremond, M. Thonnat. "Video Understanding for Complex Activity Recognition". *Machine Vision and Applications, Special Issue Paper*, Springer-Verlag, vol. 18, pp. 167-188, 2007.
- [29] D. Fleet et Y. Weiss. "Optical Flow Estimation". *Handbook of Mathematical Models in Computer Vision*, pp. 239-258, 2005.
- [30] J. Gao, A. G. Hauptmann, A. Bharucha, H. Wactlar. "Dinig Activity Analysis Using a Hidden Markov Model". *17th International Conference on Pattern Recognition (ICPR'04)*, vol. 2, pp. 915-918, Cambridge, United Kingdom, Aug. 23-26, 2004.
- [31] I. Haritaoglu, D. Harwood, L. Davis. "W⁴: Real-Time Surveillance of People and Their Activities". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp.809-830, Aug. 2000.
- [32] C. Harris et M. Stephens. "A Combined Corner and Edge Detector". *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, Manchester, 1988.
- [33] R. Hartley et A. Zisserman. "Multiple View Geomerty in Computer Vision". *Cambridge University Press*, 2nd edition, 2003.
- [34] M. Heikkilä, M. Pietikäinen. "A Texture-Based Method For Modeling the Background and Detecting Moving Objects". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no.4, pp. 657-662, Apr. 2006.
- [35] H. Hung. "Vidéosurveillance pour Appartements Intelligents : Application à la Détection de Prise de Médicaments (Smart Home : Application to the Detection of Medication Intake)". *Thèse de Doctorat, École doctorale de mathématiques et informatique*. E.d. 184 (Marseille), France 2010.

- [36] M. Isard et A. Black. "Condensation – Conditional Density Propagation for Visual Tracking". *International Journal of Computer Vision*, vol. 29, no 1, pp. 5-28, 1998.
- [37] R. Kalman. "A New Approach to Linear Filtering and Prediction Problems". *Transactions of the ASME–Journal of Basic Engineering*, 82 (Series D), pp. 35-45, 1960.
- [38] T. Lee et A. Mihailidis. "An Intelligent Emergency Response System: Preliminary Development and Testing of Automated Fall Detection". *Journal of Telemedicine and Telecare*, vol. 11, no 4, pp. 194-198, 2005.
- [39] Leignel C. et JM. Jolion. "Projet ANR CAnADA : Comportements Anormaux Analyse Détection Alerte". *Laboratoire d'Informatique en Image et Systèmes d'information*. UMR 5205 CNRS/INSA de Lyon/Université de Lyon, 2008.
- [40] X. Liu, CS. Chua. "Multi-agent Activity Recognition Using Observation Decomposed Hidden Markov Models". *Image and Vision Computing* vol. 24 pp. 166–175, 2006.
- [41] D. Lowe. "Object Recognition From Local Scale-Invariant Features". *Proceedings of the 7th International Conference on Computer Vision*, vol. 2, pp. 1150–1157, Sep. 1999.
- [42] D. Lowe. "Distinctive Image Features From Scale-Invariant Keypoints". *Proceedings of the International Conference on Computer Vision*, vol. 60, no 2 pp. 91-110, 2004.
- [43] D. Marr et T. Poggio. "A Computational Theory of Human Stereo Vision". *Royal Society*, London B 204, 1979.

- [44] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld et H. Wechsler. "Tracking Groups of People". *Computer Vision and Image Understanding*, vol. 80, pp. 42–56, 2000.
- [45] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir et L. Van Gool. "A Comparison of Affine Region Detectors". *International Journal of Computer Vision*, vol. 65, pp. 43-72, 2005.
- [46] H. Moravec. "Rover Visual Obstacle Avoidance". *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pp. 785-790, Vancouver, Colombie Britannique, Aug. 1981.
- [47] N. Noury, P. Rumeau, A. Bourke, G. OLaighin et J.E. Lundy. "A proposal for the Classification and Evaluation of Fall Detectors". *IRBM*, vol. 29, no 6, pp. 340-349, Dec. 2008.
- [48] M. Piccardi. "Background Subtraction Techniques: a Review". *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics*, pp. 3099–3104. The Hague, Netherlands, Oct. 2004.
- [49] L. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". *IEEE Proceedings*, vol. 77, no 2, pp. 257–286, 1989.
- [50] C. Rougier, J. Meunier, A. St-Arnaud et J. Rousseau, "Robust Video Surveillance for Fall Detection". *IEEE Transaction Circuits and Systems for Video Technology*, pp. 611 – 622, May 2011.
- [51] C. Rougier, J. Meunier, A. St-Arnaud et J. Rousseau. " 3D Head Trajectory Using a Single Camera". *International Journal of Future Generation Communication and Networking, special issue "Image and Signal Processing"*, vol. 3, no. 4, pp. 43-54, Dec. 2010 (papier invité).

- [52] C. Rougier. "Vidéosurveillance Intelligente pour la Détection des Chutes chez les Personnes Âgées". *Thèse de Ph.D en informatique*, Université de Montréal, Mar. 2010.
- [53] A. de Saint Vincent. "Perception et Modélisation de l'Environnement d'un Robot Mobile: Une Approche par Stéréovision". *PhD thesis*, LAAS, Toulouse, France, 1986.
- [54] S. Sarkar, J. Phillips, Z. Liu, I. Robledo Vega, P. Grother et K. Bowyer. "The HumanID Gait Challenge Problem: Data Sets, Performance, and Analysis". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 162-177, Feb. 2005.
- [55] N. Siebel et S. Maybank. "The ADVISOR Visual Surveillance System". *Proceedings of the ECCV 2004 Workshop, Applications of Computer Vision (ACV'04)*, ISBN 80-01-02977-8, Prague, Czech Republic, pp. 103-111, May 2004.
- [56] T. Skordas et R. Horaud. "Mise en Correspondance de Structures 2D dans une Paire d'Images Stéréoscopiques". *Rapport de recherche INRIA, 2^{ème} congrès TIPI*, (Savoie, France), 1989.
- [57] P. Sirinivasan, G. Qian, D. Birchfield et A. Kidané. "Design of a Pressure Sensitive Floor for Multimodal Sensing". *Proceeding of the 9th International Conference on Information Visualisation*, pp. 41-46, Jul. 2005.
- [58] C. Stauffer et W. Grimson. "Adaptative Background Mixture Models for Real Time Tracking", *IEEE, Computer Vision and Pattern Recognition*, vol. 2, pp. 246-252, 1999.
- [59] R. Szeliski. "Computer Vision: Algorithms and Applications". Springer. Sep. 2010.

- [60] M. Valin, J. Meunier, A. St Arnaud et J. Rousseau. "Video Surveillance of Medication Intake". *Proceedings of the 28th IEEE. EMBS Annual International Conference*, pp. 6393-6399, New York City, USA, Aug. 30-Sep. 3, 2006.
- [61] P. Viola, M. Jones et D. Snow. "Detecting Pedestrians Using Patterns of Motion and Appearance". *International Journal of Computer Vision*, vol. 63, no 2, pp. 153–161, 2005.
- [62] C. Xu, J. Liu et B. Kuipers. "Motion Segmentation by Learning Homography Matrices from Motor Signals". *Computer and Robot Vision*, pp. 316-323, St. Johns, NewFoundLand, Canada, 2011.
- [63] A. Yilmaz, O. Javed et M. Shah. "Object Tracking: A Survey". *ACM Computing Surveys*, vol. 38, no 4, Article 13, Dec. 2006.
- [64] M. Zia Uddin, T. Kim, J. Tai Kim. "Video-Based Human Gait Recognition Using Depth Imaging and Hidden Markov Model : A Smart System for Smart Home". SHB2010 – 3rd *International Symposium on Sustainable Healthy Buildings*, pp.153-179, Seoul, Korea. 27 May 2010.
- [65] Y. Zigel, D. Litvak et I. Gannot. "A Method for Automatic Fall Detection of Elderly People Using Floor Vibrations and Sound – Proof of Concept on Human Mimicking Doll Falls". *IEEE Transactions on Biomedical Engineering*, vol. 56, no 12, pp. 2858-2867, 2009